Fundamentals of Multimedia, 2nd ed.

ISBN: 978-3-319-05289-2
Electronic ISBN: 978-3-319-05290-8
DOI: 10.1007/978-3-319-05290-8_1

Ze-Nian Li, Mark S. Drew, and Jiangchuan Liu

School of Computing Science

Simon Fraser University

Vancouver, Canada

Exercise Solutions

(Preliminary Version)

Contents

Chapter 1

Introduction to Multimedia

Exercises

1. Using your own words, describe what is ”multimedia” ? Is multimedia simply a collection of different types of media?

2. Identify three novel multimedia applications. Discuss why you think these are novel and their potential impact.

3. Discuss the relation between multimedia and hypermedia.

4. Briefly explain, in your own words, “Memex” and its role regarding hypertext. Could we carry out the Memex task today? How do you use Memex ideas in your own work?

   Answer:
   Memex was a theoretical system explicated by Vanevar Bush in a famous 1945 essay. His main ideas involved using *associative memory* as an aid for organizing a welter of material. He even adumbrated the concept of *links*.

5. Discover a current media input, storage, or playback device that is analog. Is it necessary to convert to digital? What are the pros and cons to be analog or digital?

6. Your task is to think about the transmission of smell over the Internet. Suppose we have a smell sensor at one location and wish to transmit the *Aroma Vector* (say) to a receiver to reproduce the same sensation. You are asked to design such a system. List three key issues to consider and two applications of such a delivery system. *Hint*: Think about medical applications.

   Answer:
   Digital scent technology may actually be just around the corner – if consumers actually want such a thing: they have voted with their investments in the past by ignoring such companies. On the other hand, doctors do rely on scent for valuable input, so perhaps this could be used in telemedicine.

7. Tracking objects or people can be done by both sight and sound. While vision systems are precise, they are relatively expensive; on the other hand, a pair of microphones can detect a person’s *bearing* inaccurately but cheaply. Sensor *fusion* of sound and vision is thus useful. Surf the web to find out who is developing tools for video conferencing using this kind of multimedia idea. Distributed Meetings: A Meeting Capture and Broadcasting System, and PING: A Group-to-Individual Distributed Meeting System are under development by Microsoft Research.

8. *Non-photorealistic* graphics means computer graphics that do well enough without attempting to make images that look like camera images. An example is conferencing. For example, if we track lip movements, we can generate the right animation to fit our face. If we don't much like our own face, we can substitute another one — facial-feature modeling can map correct lip movements onto another model. See if you can find out who is carrying out research on generating avatars to represent conference participants' bodies.

   Answer:
   Non-Photorealistic Animation and Rendering is an ongoing research topic, last considered in detail at a International Symposium co-lated with SIGGRAPH 2013.

9. Watermarking is a means of embedding a hidden message in data. This could have important legal implications: Is this image copied? Is this image doctored? Who took it? Where? Think of "messages" that could be sensed while capturing an image and secretly embedded in the image, so as to answer these questions. (A similar question derives from the use of cell phones. What could we use to determine who is putting this phone to use, and where, and when? This could eliminate the need for passwords or others using the phone you lost.)

   Answer:
   Embed retinal scan plus date/time, plus GPS data; sense fingerprint.

Chapter 2

A Taste of Multimedia

Exercises

1. What extra information is multimedia good at conveying?

   (a) What can spoken text convey that written text cannot?
   Answer:
   Speed, rhythm, pitch, pauses, etc...
   Emotion, feeling, attitude ...

   (b) When might written text be better than spoken text?
   Answer:
   Random access, user-controlled pace of access (i.e. reading vs. listening)
   Visual aspects of presentation (headings, indents, fonts, etc. can convey information)
   For example: the following two pieces of text may sound the same when spoken:
   I said "quickly, come here."
   I said quickly "come here."

2. Find and learn Autodesk 3ds Max (formerly 3D Studio Max) in your local lab software. Read the online tutorials to see this software's approach to a 3D modeling technique. Learn texture mapping and animation using this product. Make a 3D model after carrying out these steps.

3. Design an interactive web page using Adobe Dreamweaver. HTML 4 provides *layer* functionality, as in Adobe Photoshop. Each layer represents an HTML object, such as text, an image, or a simple HTML page (and the Adobe HTML5 Pack is an extension to Adobe Dreamweaver). In Dreamweaver, each layer has a marker associated with it. Therefore, highlighting the layer marker selects the entire layer, to which you can apply any desired effect. As in Flash, you can add buttons and behaviors for navigation and control. You can create animations using the Timeline behavior.

4. Suppose we wish to create a simple animation, as in Figure 2.1. Note that this image is exactly what the animation looks like at some time, not a figurative representation of the *process* of moving the fish; the fish is repeated as it moves. State what we need to carry out this objective, and give a simple pseudocode solution for the problem. Assume we already have a list of (x, y) coordinates for the fish path, that we have available a procedure for centering images on path positions, and that the movement takes place on top of a video.

   Answer:

```
 \\ We have a fish mask as in Figure \ref{FIG:MASKANDSPRITE}(a), and
 \\ also a fish sprite as in Figure \ref{FIG:MASKANDSPRITE}(b).
 \\Apply sprites as in text.
% \\ Fish positions have centers posn(t).x posn(t).y
%
% currentmask = an all-white image
% currentsprite = an all-black image
% for t = 1 to maxtime {
%    \\ Make a mask fishmask with the fish mask black area
%    \\ centered on position posn(t).x, posn(t).y
%    \\ and a sprite fishsprite with the colored area also moved
%    \\ to posn(t).x, posn(t).y
%    \\ Then expand the mask:
%    currentmask = currentmask AND fishmask \\ enlarges mask
%    currentsprite = currentsprite OR fishsprite \\ enlarges sprite
%    \\ display current frame of video with fish path on top:
%    currentframe = (frame(t) AND currentmask) OR currentsprite
% }
%
```

5. For the slide transition in Figure 2.8, explain how we arrive at the formula for x in the unmoving right video R_R.

   Answer:
   if $x/x_{max} \geq t/t_{max}$, then we are in the right-hand video.
   The rest of the task is left up to the reader.

6. Suppose we wish to create a video transition such that the second video appears under the first video through an opening circle (like a camera iris opening), as in Figure 2.22. Write a formula to use the correct pixels from the two videos to achieve this special effect. Just write your answer for the red channel.

   Answer:

y ˆ

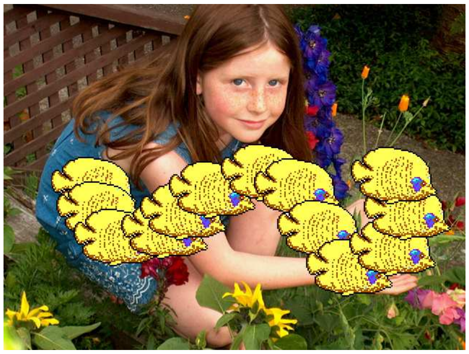
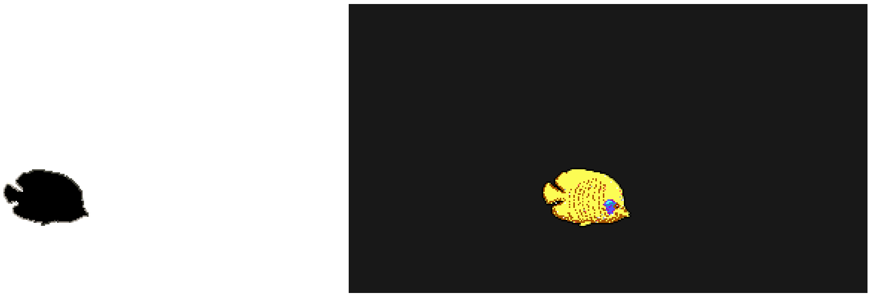
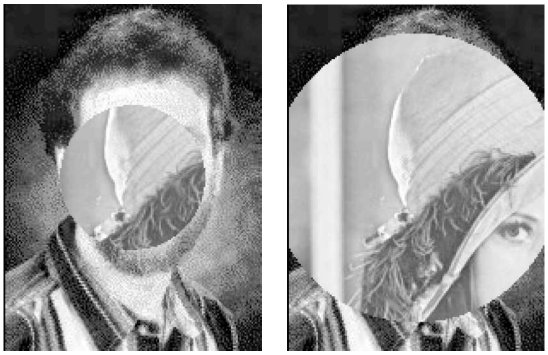


Fig. 2.1: Sprite, progressively taking up more space.



(a) (b)

Fig. 2.30: (answer) Mask and Sprite.



(a) (b)

Fig. 2.31: Iris wipe: (a) iris is opening; (b) at a later moment.

```
| | R0              |
| |                 |
| |    _____        |
| |   (     )       |
| |   ( R1  )       |
| |   (_____)       |
| |                 |
| |                 |
|  ------------------

   -----------------> x
```

```
At x,y,
  radius = sqrt(  (x-x_max/2)^2 + (y-y_max/2)^2  )

  If (  r < (t/t_max)*r_max  )
     R(x,y,t) = R1(x,y,t)
  Else
     R(x,y,t) = R0(x,y,t)
```

7. Now suppose we wish to create a video transition such that the second video appears under the first video through a moving radius (like a clock hand), as in Figure 2.23. Write a formula to use the correct pixels from the two videos to achieve this special effect for the red channel.

Answer:

```
y ^    ____________________
  | | R0         /      |
  | |           /       |
  | |     _____ /       |
  | |    (     )        |
  | |    ( R1___________|
  | |    (_____)        |
  | |                   |
  | |                   |
  |   -----------------

      ----------------> x
```

```
  At x,y,
    angle = atan(  -(y-y_max/2)/(x-x_max/2)  )

    If (  angle < (t/t_max)*a_max  )
       R(x,y,t) = R1(x,y,t)
    Else
       R(x,y,t) = R0(x,y,t)
```

8. Suppose you wish to create a wavy effect, as in Figure 2.24. This effect comes from replacing the image x value by an x value offset by a small amount. Suppose the image size is 160 rows × 120 columns of pixels.

   (a) Using float arithmetic, add a sine component to the x value of the pixel such that the pixel takes on an RGB value equal to that of a different pixel in the original image. Make the maximum shift in x equal to 16 pixels.

   Answer:
   R = R(x + sin(y/120) * 16 , y) and similarly for G, B.

   (b) In Premiere and other packages, only integer arithmetic is provided. Functions such as `sin` are redefined so as to take an `int` argument and return an `int`. The argument to the `sin` function must be in $0 .. 1{,}024$, and the value of `sin` is in $-512 .. 512$: `sin(0)` returns `0`, `sin(256)` returns `512`, `sin(512)` returns `0`, `sin(768)` returns `-512` and `sin(1,024)` returns `0`.

   Rewrite your expression in part (a) using integer arithmetic.

   Answer:

   ```
   R = R(x + sin( (y*1024)/120 ) /32,y) and similarly
   for G,B.
   ```

   (c) How could you change your answer to make the waving time-dependent?

   Answer:

```
R = R(x +  t*sin(y*(1024/120) ) /(32*tmax),y)
```

9. How would you create the color wheel image in Figure 2.3? Write a small program to make such an image. *Hint:* Place R, G, and B at the corners of an equilateral triangle inside the circle. It's best to go over *all columns and rows* in the output image rather than simply going around the disk and trying to map results back to (x, y) pixel positions.

   Answer:

```
SIZE = 256;
im = ones(SIZE,SIZE,3);
Place R at (0,1).
Place G at 120 degrees.
Place B at 240 degrees.
The outside perimeter goes from R to G as we go from
  R to G.
And from B to R as we go from 240 to 360.

At a position where the Outside Perimeter value
is   out  , at radius   r  the color is
(1-r)*(1,1,1) + r*(out)
```

10. As a longer exercise for learning existing software for manipulating images, video, and music, make a 1-minute digital video. By the end of this exercise, you should be familiar with PC- or Apple-based equipment and know how to use a video editor (e.g., Adobe Premiere), an image editor (especially Photoshop), some music notation program for producing MIDI, and perhaps digital-audio manipulation software such as Adobe Audition, as well as other multimedia software.

    (a) Acquire (or find) at least three digital video files. You can either use a camcorder or download some from the net, or use the video setting on still-image camera, phone, etc. (or, for interesting legacy video, use video-capture through Premiere or an equivalent product to make your own, from an old analog Camcorder or VCR — this is challenging, and fun).

    (b) Try to upload one of the videos to YouTube. Check the time that is taken to upload the video, and discuss its relation with your video's quality and size. Is this time longer or shorter than the total playback time of the video?

    (c) Compose (or edit) a small MIDI file with music-manipulation software.

    (d) Create (or find) at least one WAV file (ok – could be MP3). You may either digitize your own or find some on the net, etc. You might like to edit this digital-audio file using software such as Audition, Audacity, etc.

    (e) Use Photoshop to create a title and an ending. This is not trivial; however, you cannot say you know about multimedia without having worked with Photoshop.

    A useful feature to know in Photoshop is how to create an alpha channel:

    - Use an image you like: a .JPG, say.
    - Make the background some solid color, white, say.
    - Make sure that you have chosen `Image > Mode > RGB Color`.
    - Select that background area (you want it to remain opaque in Premiere): MagicWandTool

- `Select > Save Selection > Channel=New; OK`
- `Window > ShowChannels`; Double click the new channel and rename it Alpha; make its color (0,0,0)
- Save the file as a .PSD

If the alpha channel you created in Photoshop has a white background, you'll need to choose ReverseKey in Premiere when you choose `Transparency > Alpha`.

(f) Combine all of the above to produce a movie about 60 seconds long, including a title, some credits, some soundtracks, and at least three transitions. The plotline of your video should be interesting, to you!

(g) Experiment with different compression methods; you are encouraged to use MPEG for your final product. We are very interested in seeing how the concepts in the textbook go over into the production of actual video. Adobe Premiere can use the DivX codec to generate movies, with the output movie actually playable on (that) machine; but wouldn't it be interesting to try various codecs?

(h) The above constitutes a minimum statement of the exercise. You may be tempted to get very creative, and that's fine, but don't go overboard and take too much time away from the rest of your life!

Chapter 3

Graphics and Image Data Representations

Exercises

1. Briefly explain why we need to be able to have less than 24-bit color and why this makes for a problem. Generally, what do we need to do to adaptively transform 24-bit color values to 8-bit ones?

   Answer:

   Large file sizes; or not 24-bit display.
   Cluster color pixels so as to best use the bits available.

2. Suppose we decide to quantize an 8-bit grayscale image down to just 2 bits of accuracy. What is the simplest way to do so? What ranges of byte values in the original image are mapped to what quantized values?

   Answer:
   Just use the first 2 bits in the grayscale value.

3. Suppose we have a 5-bit grayscale image. What size of ordered dither matrix do we need to display the image on a 1-bit printer?

   Answer:

   ```
    2^5=32 levels ~= n^2+1 with n=6; therefore need D(6)
   ```

4. Suppose we have available 24 bits per pixel for a color image. However, we notice that humans are more sensitive to R and G than to B — in fact, 1.5 times more sensitive to R or G than to B. How could we best make use of the bits available? How could we best make use of the bits available?

   Answer:
   ratio is 3:3:2, so use bits 9:9:6 for R:G:B.

5. At your job, you have decided to impress the boss by using up more disk space for the company's grayscale images. Instead of using 8 bits per pixel, you'd like to use 48 bits per pixel in RGB. How could you store the original grayscale images so that in the new format they would appear the same as they used to, visually?

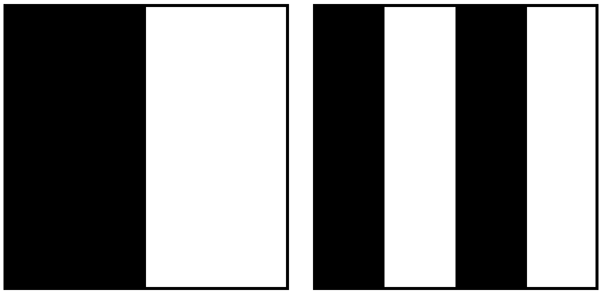
Answer:
48 bits RGB means 16 bits per channel: so re-store the old ints, which were $< 2^8$, as new ints $< 2^{16}$. But then multiply the old values by 2^8.

6. Suppose an 8-bit greyscale image appears as in Fig. 3.19(a); i.e., linear shading goes from 0 to 255 from left to right, illustrated in Fig. 3.19(b).
   The image is 100 rows by 100 columns.

   For the most significant bitplane, please draw an image showing the 1's and 0's.
   How many 1's are there?

   For the next-most significant bitplane, please draw an image showing the 1's and 0's.
   How many 1's are there?

   Answer:



(a) (b)

Fig. 3.1: (a): MSB=bit_0; (b): bit_1

7. For the color LUT problem, try out the median-cut algorithm on a sample image. Explain briefly why it is that this algorithm, carried out on an image of red apples, puts more color gradation in the resulting 24-bit color image where it is needed, among the reds.

8. In regard to nonordered dithering, a standard graphics text [2] states, "Even larger patterns can be used, but the spatial versus intensity resolution trade-off is limited by our visual acuity (about one minute of arc in normal lighting)."

   (a) What does this sentence mean?
   Answer:
   The larger is the matrix of patterns, the greater is the possibility that we can see gaps between dots.

   (b) If we hold a piece of paper out at a distance of 1 foot, what is the approximate linear distance between dots? (*Information*: One minute of arc is 1/60 of one degree of angle. Arc length on a circle equals angle (in radians) times radius.) Could we see the gap between dots on a 300 dpi printer?
   Answer:
   One minute of arc is $1/60 * \pi/180$ radians, and at $r = 25$ cm, the arc length is $25 * \pi/(60 * 180)$. For a 300 dpi printer, we could just see such a gap.

9. Write down an algorithm (pseudocode) for calculating a color histogram for RGB data.

   Answer:

```
for i=0..(MAX_Y-1)
  for j=0..(MAX_X-1)

     R = image[x][y].red;
     G = image[x][y].green;
     B = image[x][y].blue;
     hist[R][G][B]++;
```

10. **Describe in detail how to use a *single* image and several color lookup tables to realize a simple animation — a rotating color wheel, in which a sequence of 4 snapshots will appear repetitively. The wheel rotates $90°$ clockwise each time.**

Chapter 4

Color in Image and Video

Exercises

1. Consider the following set of color-related terms:

   (a) wavelength

   (b) color level

   (c) brightness

   (d) whiteness

   How would you match each of the following (more vaguely stated) characteristics to each of the above terms?

   (a) luminance ⇒ **brightness**

   (b) hue ⇒ **wavelength**

   (c) saturation ⇒ **whiteness**

   (d) chrominance ⇒ **color level**

2. What color is outdoor light? I.e., around what wavelength would you guess the peak power is for a red sunset? For blue sky light?

   Answer:
   450 nm, 650 nm.

3. "The LAB gamut covers all colors in the visible spectrum."

   What does that statement mean? Briefly, how does LAB relate to color? – just be descriptive.

   What are (roughly) the relative sizes of the LAB gamut, the CMYK gamut, and a monitor gamut?

   Answer:
   CIELAB is simply a (nonlinear) restating of XYZ tristimulus values. The objective of CIELAB is to develop a more perceptually uniform set of values, for which equal distances in different parts of gamut imply roughly equal differences in perceived color.

   XYZ, or equivalently CIELAB, by definition covers the whole human visual system gamut. In comparison, a monitor gamut covers just the triangle joining the R, G, and B pure-phosphor-color corners, so is much smaller. Usually, a printer gamut is smaller again.

4. Prove that straight lines in (X, Y, Z) space project to straight lines in (x, y) chromaticity space. I.e., let $C_1 = (X_1, Y_1, Z_1)$ and $C_2 = (X2, Y2, Z2)$ be two different colors, and let $C_3 = (X_3, Y_3, Z_3)$ fall on a line connecting C_1 and C_2: $C_3 = \alpha C_1 + (1 - \alpha)C_2$. Then show that $(x_3, y_3) = \beta(x_1, y_1) + (1 - \beta)(x_2, y_2)$ for some β.

5. Where does the chromaticity "horseshoe" shape Figure 4.11 come from? Can we calculate it? Write a small pseudocode solution for the problem of finding this so-called "spectrum locus".

   Hint: Fig. 4.20(a) shows the color-matching functions in Fig. 4.10 drawn as a set of points in 3-space. And Fig. 4.20(b) shows these points mapped into another 3D set of points.
   Hint: Try a programming solution for this problem, to help you answer it more explicitly.

   Answer:
   The $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$, color-matching curves define the human visual system response to spectra. The outer boundary corresponds to the "spectrum locus", i.e., the response to a pure (laser-like) single wavelength sample. Interior points on the surface correspond to mixtures of wavelengths. To make visualization simpler, we go to a 2D chromaticity color space: $\{x, y\} = \{X, Y\}/(X + Y + Z)$. The boundary of the XYZ plot, projected to 2D, resembles a horseshoe shape.

6. **Suppose we use a new set of color–matching functions $\bar{x}^{new}(\lambda)$, $\bar{y}^{new}(\lambda)$, $\bar{z}^{new}(\lambda)$ with values**

λ (nm)	$\bar{x}^{new}(\lambda)$	$\bar{y}^{new}(\lambda)$	$\bar{z}^{new}(\lambda)$
450	0.2	0.1	0.5
500	0.1	0.4	0.3
600	0.1	0.4	0.2
700	0.6	0.1	0.0

   In this system, what are the chromaticity values (x, y) of equi-energy white light $E(\lambda)$ where $E(\lambda) \equiv 1$ for all wavelengths λ? Explain.

   Answer:
   The chromaticity values (x, y) are made from the XYZ triple $X = \sum_\lambda[\bar{x}(\lambda) \star E(\lambda)]$, $Y = \sum_\lambda[\bar{y}(\lambda) \star E(\lambda)]$, $Z = \sum_\lambda[\bar{z}(\lambda) \star E(\lambda)]$. For the new color-matching functions, since every $E(\lambda)$ is 1 for equi-energy white light, we form X, Y, Z via $\sum(\bar{x}), \sum(\bar{y}), \sum(\bar{z})$ = $(1, 1, 1)$, according to the values in the table; so the chromaticity is $x = X/(X + Y + Z)$ = $1/3$, and also $y = 1/3$.

7. Repeat the steps leading up to eq.(4.18), but this time using the NTSC standard — if you use the number of significant digits as in Table 4.1 you will end up with the transform in eq.(4.19).

8. (a) Suppose images are *not* gamma-corrected by a camcorder. Generally, how would they appear on a screen?

   Answer:
   Too dark at the low-intensity end.

   (b) What happens if we artificially increase the output gamma for stored image pixels? (One can do this in Photoshop.) What is the effect on the image?

   Answer:
   Increase the number of bright pixels — we increase the number of pixels that map to the upper half of the output range. This creates a lighter image. – and incidentally, we also decrease highlight contrast and increase contrast in the shadows.

9. Suppose image file values are in $0..255$ in each color channel. If we define $\overline{R} = R/255$ for the Red channel, we wish to carry out gamma correction by passing a new value $\overline{R}'$ to the display device, with $\overline{R}' \simeq \overline{R}^{1/2.0}$.

   It is common to carry out this operation using integer math. Suppose we approximate the calculation as creating new integer values in $0..255$ via

   $$(int)\,(255 \cdot (\overline{R}^{1/2.0}))$$

   (a) Comment (very roughly) on the effect of this operation on the number of actually available levels for display.
   Hint: coding this up in any language will help you understand the mechanism at work better – and as well, then you can simply count the outout levels.

   (b) Which end of the levels $0..255$ is affected most by gamma correction, the low end near 0 or the high end near 255? Why? How much at each end?

   Answer:

   (a) The integer values actually taken on are not as many as 256. (the number of levels comes out to 193).

   (b) At the low end, the integer value $R = 0$ **corresponds to the quantized value 0, whereas the int value 1 corresponds to**

   ```
   (int) ( 255* sqrt( 1/255 ) ) ˜= (int) ( 255/16 ) = 15
   ```

   so that an image value of 1 becomes the gamma-corrected value 15.

   At the high end, $255 \mapsto 255$ **correctly, but** $254 \;\mapsto\; 255 * (1 - 1/255) \wedge 0.5 \;\simeq\; 255 * (1 - 0.5 * 1/255) \;=\; 255 - 0.5 \;\simeq\; 254$**, so that the high end is much less affected.**

   Therefore overall the gamma-quantization greatly reduces the resolution of quantization at the low end of image intensity.

10. In many Computer Graphics applications, γ-correction is performed only in the color LUT (look-up table). Show the first 5 entries of the color LUT if it meant for use in γ-correction.
    Hint: coding this up saves you the trouble of using a calculator for this question.

    Answer:
    Use `round` **instead, but the idea is as in last question, applied now to a LUT.** $V' = V^{\frac{1}{\gamma}}$
    For answers: Do a LUT table for R G B, 256 rows, say.
    First row, indexed by 0, all entries 0. Any other kth row,
    $R = G = B = round(255 * (k/255)^{\frac{1}{\gamma}})$
    First five: 0 16 23 28 32

11. Devise a program to produce Fig. 4.1, showing the color gamut of a monitor that adheres to SMPTE specifications.

    Answer:
    The simplest answer is derived from a Fortran program on the net:
    http://www.physics.sfasu.edu/astro/color/chromaticity.html

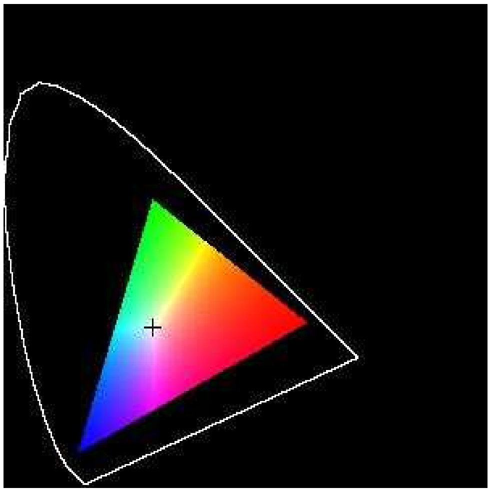


Fig. 4.1: SMPTE Monitor Gamut

12. The “hue” is the color, independent of brightness and how much pure white has been added to it. We can make a *simple* definition of hue as the set of ratios R:G:B.

    (a) Suppose a color (i.e., an RGB) is divided by 2.0, so that the RGB triple now has values 0.5 times its former values.

    Explain using numerical values:

    i. If gamma-correction is applied after the division by 2.0 and before the color is stored, does the darker RGB have the same hue as the original in the sense of having the same ratios R:G:B of light emanating from the display device? (we’re not discussing any psychophysical effects that change our perception – here we’re just worried about the machine itself).

    ii. If gamma-correction is *not* applied, does the second RGB above, = RGB/2, have the same hue as the first RGB, when displayed? And are these the same hues as for the original color as *stored*, not the light as displayed?

    (b) Assuming no gamma-correction is applied, For what color triples is the hue as displayed the same as for the original color as stored?

    Answer:

    (a) With gamma correction, RGB is stored as (RGB)∧(1/gamma) and (RGB/2) is stored as (RGB/2)∧(1/gamma). After the gamma takes effect, color∧gamma, the gamma-correction power law is reversed, and we’re back to RGB and RGB/2, so the hue does not change.

    (b) But if there is no gamma-correction, then RGB results in light we see a different hue. Example: RGB=1,1/2,1/4.

    (c) Any color with some equal entries, e.g. (1,1,0) will be the same hue, just darker. And any color with two ”0” entries will also be the same hue, just darker.

13. We wish to produce a graphic that is pleasing and easily readable. Suppose we make the background color `pink`. What color text font should we use to make the text most readable? Justify your answer.

    Answer:

```
Pink is a mixture of white and red;
Then complementary color is (1,1,1)-pink,
which is pale cyan.
```

14. To makes matters simpler for eventual printing, we buy a camera equipped with CMY sensors, as opposed to RGB sensors (CMY cameras are in fact available).

    (a) Draw spectral curves roughly depicting what such a camera's sensitivity to wavelength might look like.

    Answer:

    (see plot Fig. 4.2)

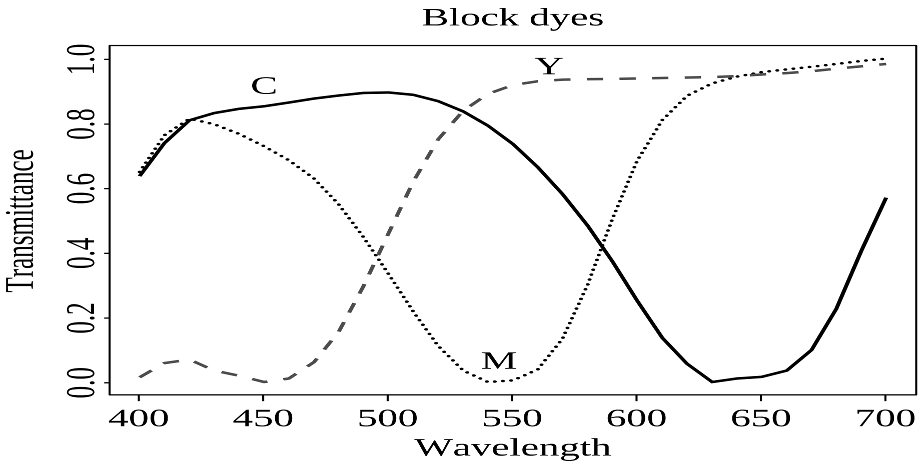


Fig. 4.2: CMY "Block" dye transmittance.

    (b) Could the output of a CMY camera be used to produce ordinary RGB pictures? How?

    Answer:

    Suppose we had C=G+B (i.e., with coefficients 1.0 times G and B), etc. Then C=R,G,B - R,0,0 = 1-R, and so R=1-C. So use R=1-C, G=1-M, B=1-Y.

15. Color ink-jet printers use the CMYK model. When the color ink *cyan* is sprayed onto a sheet of white paper,
    (i) why does it look cyan under daylight?
    (ii) what color would it appear to be under a *blue* light. Why?

    Answer:
    (i) RED from the daylight is absorbed (subtracted).
    (ii) BLUE. The CYAN ink will not absorb BLUE, and BLUE is the only color in the light.

Chapter 5

Fundamental Concepts in Video

Exercises

1. NTSC video has 525 lines per frame and 63.6 μsec per line, with 20 lines per field of vertical retrace and 10.9 μsec horizontal retrace.

   (a) Where does the 63.6 μsec come from?

   Answer:

   1 / (525 lines/frame$\times$29.97 frame/sec) = 63.6$\times 10^{-6}$ sec/line

   (b) Which takes more time, horizontal retrace or vertical retrace? How much more time?

   Answer:

   horiz = 10.9$\times 10^{-6}$ sec,
   vertical is 20 line * 63.6 μsec = 1272 μsec = 1.272 msec, so
   vertical is 1272/10.9 = 117 times longer than horizontal.

2. Which do you think has less detectable flicker, PAL in Europe or NTSC is North America? Justify your conclusion.

   Answer:
   PAL could be better since more lines, but is worse because of fewer frames/sec.

3. Sometimes the signals for television are combined into fewer than all the parts required for TV transmission.

   (a) Altogether, how many and what are the signals used for studio broadcast TV?

   Answer:
   5
   R, G, B, audio, sync; can say "blanking" instead, too.

   (b) How many and what signals are used in S-Video? What does S-Video stand for?

   Answer:
   Luminance+chrominance = 2+audio+sync = 4
   Separated video

   (c) How many signals are actually broadcast for standard analog TV reception? What kind of video is that called?

Answer:
1
Composite

4. Show how the Q signal can be extracted from the NTSC chroma signal C (Eq. 5.1) during the demodulation process.

   Answer:

   To extract Q:

   (a) Multiply the signal C by $2\sin(F_{sc}t)$, i.e.,

$$\begin{aligned} C \cdot 2\sin(F_{sc}t) &= I \cdot 2\sin(F_{sc}t)\cos(F_{sc}t) + Q \cdot 2\sin^2(F_{sc}t) \\ &= I \cdot \sin(2F_{sc}t) + Q \cdot (1 - \cos(2F_{sc}t)) \\ &= Q + I \cdot \sin(2F_{sc}t) - Q \cdot \cos(2F_{sc}t). \end{aligned}$$

   (b) Apply a low-pass filter to obtain Q and discard the two higher frequency ($2F_{sc}$) terms.

5. One sometimes hears that the old Betamax format for videotape, which competed with VHS and lost, was actually a better format. How would such a statement be justified?

   Answer:
   Betamax has more samples per line: 500, as opposed to 240.

6. We don't see flicker on a workstation screen when displaying video at NTSC frame rate. Why do you think this might be?

   Answer:
   NTSC video is displayed at 30 frames per sec, so flicker is possibly present. Nonetheless, when video is displayed on a workstation screen the video buffer is read and then rendered on the screen at a much higher rate, typically the refresh rate — 60 to 90 Hz — so no flicker is perceived. (And in fact most display systems have double buffers, completely removing flicker: since main memory is much faster than video memory, keep a copy of the screen in main memory and then when we this buffer update is complete, the whole buffer is copied to the video buffer.)

7. Digital video uses *chroma subsampling*. What is the purpose of this? Why is it feasible?

   Answer:
   Human vision has less acuity in color vision than it has in black and white — one can distinguish close black lines more easily than colored lines, which soon are perceived just a mass without texture as the lines move close to each other. Therefore, it is acceptable perceptually to remove a good deal of color information. In analog, this is accomplished in broadcast TV by simply assigning a smaller frequency bandwidth to color than to black and white information. In digital, we "decimate" the color signal by subsampling (typically, averaging nearby pixels). The purpose is to have less information to transmit or store.

8. What are the most salient differences between ordinary TV and HDTV/UHDTV?

   Answer:
   More pixels, and aspect ratio of 16/9 rather than 4/3.
   What was the main impetus for the development of HDTV/UHDTV?

Immersion — "being there". Good for interactive systems and applications such as virtual reality.

9. What is the advantage of interlaced video? What are some of its problems?

   Answer:
   Positive: Reduce flicker. Negative: Introduces serrated edges to moving objects and flickers along horizontal edges.

10. One solution that removes the problems of interlaced video is to de-interlace it. Why can we not just overlay the two fields to obtain a de-interlaced image? Suggest some simple de-interlacing algorithms that retain information from both fields.

    Answer:
    The second field is captured at a later time than the first, creating a temporal shift between the odd and even lines of the image.
    The methods used to overcome this are basically two: non-motion compensated and motion compensated de-interlacing algorithms.

    The simplest non-motion compensated algorithm is called "Weave"; it performs linear interpolation between the fields to fill in a full, "progressive", frame. A defect with this method is that moving edges show up with significant serrated lines near them.

    A better algorithm is called "Bob": in this algorithm, one field is discarded and a a full frame is interpolated from a single field. This method generates no motion artifacts (but of course detail is reduced in the resulting progressive image).

    In a vertical-temporal (VT) de-interlacer, vertical detail is reduced for higher temporal frequencies. Other, non-linear, techniques are also used.

    Motion compensated de-interlacing performs inter-field motion compensation and then combines fields so as to maximize the vertical resolution of the image.

11. Assuming the bit-depth of 12 bits, 120 fps, and 4:2:2 chroma subsamplng, what are the bitrates of the 4K UHDTV and 8K UHDTV videos if they are uncompressed?

    Answer:
    4K UHDTV: $3840 \times 2160 \times 120 \times 24 \approx 23.89$ **Gbps.**

    8K UHDTV: $7680 \times 4320 \times 120 \times 24 \approx 95.55$ **Gbps.**

12. Assuming we use the toed-in stereo camera model, the interocular distance is I, and the screen is D meters away, (a) At what distance will a point P generate a positive parallax equal to I on the screen? (b) At what distance will a point P generate a negative parallax equal to $-I$?

    Answer:
    (a) At infinity. (b) At $D/2$**.**

Chapter 6

Basics of Digital Audio

Exercises

1. We wish to develop a new Internet service, for doctors. Medical ultrasound is in the range 2-10 MHz; what should our sampling rate be chosen as?

   Answer:
   20MHz

2. My old Soundblaster card is an 8–bit card.

   (a) What is it 8 bits of?

   (b) What is the best SQNR (Signal to Quantization Noise Ratio) it can achieve?

   Answer:

   ```
   (a) Quantization levels (not sampling frequency)
   (b) Best SQNR is 1 level out of 256 possible levels.
   Calculate SQNR using largest value in dynamic range:
       SNR = 20 log_10 (255/2ˆ0 )
       ˜= 48 db
   ```

3. If a tuba is 20 dB louder than a singer's voice, what is the ratio of intensities of the tuba to the voice?

   Answer:
   100 times louder.

4. If a set of ear protectors reduces the noise level by 30 dB, how much do they reduce the intensity (the power)?

   Answer:
   A reduction in intensity of 1000.

5. It is known that a loss of audio output at both ends of the audible frequency range is inevitable due to the frequency response function of audio amplifier

   (a) If the output was 1 volt for frequencies at mid-range, after a loss of -3 dB at 18 KHz, what is the output voltage at this frequency?

(b) To compensate the loss, a listener can adjust the gain (and hence the output) at different frequencies from an equalizer. If the loss remains -3 dB and a gain through the equalizer is 6 dB at 18 KHz, what is the output voltage now?

[Hint: Assume $\log_{10} 2 = 0.3$.]

Answer:
(a) $20 \log \frac{V}{1} = -3$**;** $2 \log V = -0.3$**;** $2 \log V = -\log 2$**;** $\log(V^2) = -\log 2$**;** $V = \frac{1}{\sqrt{2}} = 0.7$ **volts**
(b) -3 + 6 = 3 dB; $V = \sqrt{2} = 1.4$ **volts**

6. Suppose the Sampling Frequency is 1.5 times the True Frequency. What is the Alias Frequency?

   Answer:
   0.5 times the True Frequency.

7. In a crowded room, we can still pick out and understand a nearby speaker's voice notwithstanding the fact that general noise levels may be high. This is what is known as the "cocktail-party effect"; how it operates is that our hearing can localize a sound source by taking advantage of the difference in phase between the two signals entering our left and right ears ("binaural auditory perception"). In mono, we could not hear our neighbor's conversation very well if the noise level were at all high.

   State how you think a karaoke machine works.
   Hint: the mix for commercial music recordings is such that the "pan" parameter is different going to the left and right channels for each instrument. That is, for an instrument, the left, or the right, channel is emphasized. How would the singer's track timing have to be recorded in order to make it easy to subtract out the sound of the singer? (And this is typically done.)

   Answer:
   For the singer, left and right is always mixed with the exact same pan. So subtract out the sound of the singer.

8. The *dynamic range* of a signal V is the ratio of the maximum to the minimum, expressed in decibels. The dynamic range expected in a signal is to some extent an expression of the signal quality. It also dictates the number of bits per sample needed in order to reduce the quantization noise down to an acceptable level; e.g., we may like to reduce the noise to at least an order of magnitude below V_{min}.

   Suppose the dynamic range for a signal is 60 dB. Can we use 10 bits for this signal? Can we use 16 bits?

   Answer:
   The range is mapped to $-2^{(N-1)} \ldots 2^{(N-1)} - 1$**.** V_{max} **is mapped to top value,** $\sim\ 2^{(N-1)}$**. In fact, whole range** V_{max} **down to** $(V_{max} - q/2)$ **is mapped to that, where** q **is the quantization interval. The largest negative signal,** $-V_{max}$ **is mapped to** $-2^{(N-1)}$**. Therefore** $q = (2 * V_{max})/(2^N)$**, since there are** 2^N **intervals.**

   The dynamic range is V_{max}/V_{min}**, where** V_{min} **is the smallest *positive* voltage we can see that is not masked by the noise. Since the dynamic range is 60 dB, we have** $20 \log_{10}(V_max/V_min) = 60$ **so** $V_{min} = V_{max}/1000$**.**
   At 10 bits, the quantization noise, equal to $q/2$**==half a quantization interval** q**, is** $q/2 = (2 * V_{max}/2^N)/2 = V_{max}/(2^{10})$**, or in other words** $V_{max}/1024$**. So this is not sufficient intensity resolution.**
   At 16 bits, the noise is $V_{max}/(2^{16}) = V_{max}/(64*1024)$**, which is more than an order of magnitude smaller than** V_{min} **so is fine.**

9. Suppose the dynamic range of speech in telephony implies a ratio V_{max}/V_{min} of about 256. Using uniform quantization, how many bits should we use to encode speech, so as to make the quantization noise at least an order of magnitude less than the smallest detectable telephonic sound?

   Answer:
   $V_{min} = V_{max}/256$**.**
   The quantization noise is $V_{max}/2^n$, if we use n bits. Therefore we need 12 bits.

10. *Perceptual nonuniformity* is a general term for describing the non-linearity of human perception, e.g., when a certain parameter of an audio signal varies, humans do not necessarily perceive the difference in proportion to the amount of change.

    (a) Briefly describe at least two types of Perceptual nonuniformities in human auditory perception.

    (b) Which one of them does A-law (or μ-law) attempt to approximate? Why could it improve the quantization?

    Answer:
    (a):
    (1) Logarithmic response to magnitude,
    (2) different sensitivity to different frequencies,
    (b): μ-law approximates the non-linear response to magnitude.

11. Suppose we mistakenly always use the 0.75 point instead of the 0.50 point in a quantization interval as the decision point, in deciding to which quantization level an analog value should be mapped. Above, we have a rough calculation of SQNR. What effect does this mistake have on the SQNR?

    Answer:
    Now SQNR becomes

    $$SQNR \; = \; 20 \log_{10} \frac{2^{N-1}}{\frac{3}{4}} \tag{6.1}$$

    or, in other words, if $N = 16$ we lose 9.54 - 6.02 = -3.52 dB.

12. State the Nyquist frequency for the following digital sample intervals. Express the result in Hertz in each case.

    (a) 1 millisecond

    (b) 0.005 seconds

    (c) 1 hour

    Answer:

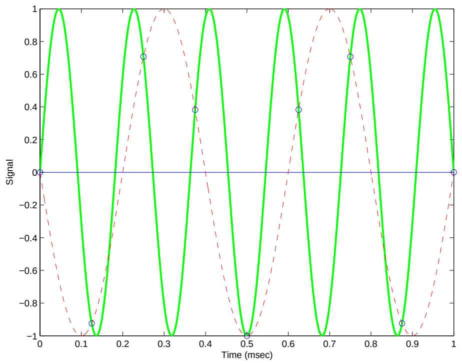
    (a) 1 millisecond → 500 Hz

    (b) 0.005 seconds → 100 Hz

    (c) 1 hour → 0.000139 Hz or 0.139 mHz

13. Draw a diagram showing a sinusoid at 5.5 kHz, and sampling at 8 kHz (just show 8 intervals between samples in your plot). Draw the alias at 2.5 kHz and show that in the 8 sample intervals, exactly 5.5 cycles of the true signal fit into 2.5 cycles of the alias signal.

    Answer:



14. In an old Western movie, we notice that a stagecoach wheel appears to be moving backwards at $5°$ per frame, even though the stagecoach is moving forward. To what is this effect due? What is the true situation?

    Answer:
    This is an effect of aliasing: the wheel is moving forward at $355°$ per frame.

15. Suppose a signal contains tones at 1 kHz, 10 kHz, and 21 kHz, and is sampled at the rate 12 kHz (and then processed with an anti-aliasing filter limiting output to 6 kHz). What tones are included in the output?
    Hint: most of the output consists of aliasing.

    Answer:
    1 kHz, 12-10=2 kHz, and 2*12-21=3 kHz tones are present.

16. The Pitch Bend opcode in MIDI is followed by two data bytes specifying how the control is to be altered. How many bits of accuracy does this amount of data correspond to? Why?

    Answer:
    14

17. (a) Can a single MIDI message produce more than one note sounding?

    Answer:
    No.

    (b) Is is possible that more than one note can be sounding on a particular instrument at once? How is that done in MIDI?

    Answer:
    Yes — use two NoteOn messages for one channel before the NoteOff message is sent.

    (c) Is the Program Change MIDI message a Channel Message? What does this message accomplish? Based on the Program Change message, how many different instruments are there in General MIDI? Why?

    Answer:
    Yes.
    Replaces patch for a channel.
    128, since has one data byte, which must be in 0..127.

    (d) In general, what are the two main kinds of MIDI messages? In terms of data, what is the main difference between the two types of messages? Within those two categories, please list the different sub-types .

Answer:
Channel Messages and System Messages.
Channel voice messages, Channel mode messages, System real-time messages, System common messages, System exclusive messages.
Channel messages have a status byte with leading most-significant-bit set, and 4 bits of channel information; System messages have the 4 MSBs set.

18. The note “A above Middle C” (with frequency 440 Hz) is note 69 in General MIDI. What MIDI bytes (in hex) should be sent to play a note twice the frequency of (i.e., one octave above) “A above Middle C” at maximum volume on channel 1? (Don’t include start/stop bits.)
Information: An octave is 12 steps on a piano, i.e., 12 notes up.

Answer:

```
 (a)  The note to play is one octave above # 69;
which is note 69+12 = 81
Channel 1 is 0, internally, so send
  &H90  &H51  &H7F
  note-on,channel-1;  note#81;  max-velocity

 (b)  Send note off right after:
  &H80  &H51  &H7F
  note-off,channel-1;  note#81;  max-velocity==value doesn't matter
```

19. Give an example (in English, not hex) of a MIDI voice message.

Answer:
NoteOn

Describe the parts of the “assembler” statement for the message you suggested above.

Answer:
(1) opcode=Note on; (2) data = note, or key, number; (3) data = ”velocity”==loudness.

What does a “program change” message do?

Answer:
Replaces a channel’s timbre with a new one; e.g. piano → violin Suppose “Program change” is hex &HC1 . What does the instruction &HC103 do?

Answer:
Changes the patch to #4 on channel 2.

20. We have suddenly invented a new kind of music: “18-tone music”, that requires a keyboard with 180 keys. How would we have to change the MIDI standard to be able to play this music?

Answer:
Need more bytes for the data part of the Note-On, Note-Off messages: if we used two bytes, instead of one, we could go from allowing up to 128 different notes to as many as 2^{14}.

21. In PCM, what is the *delay*, assuming 8 kHz sampling? Generally, delay is the penalty associated with any algorithm due to sampling, processing, and analysis.

Answer:
Since there is no processing associated with PCM, the delay is simply the time interval between two samples, and at 8 kHz, this is 0.125 msec.

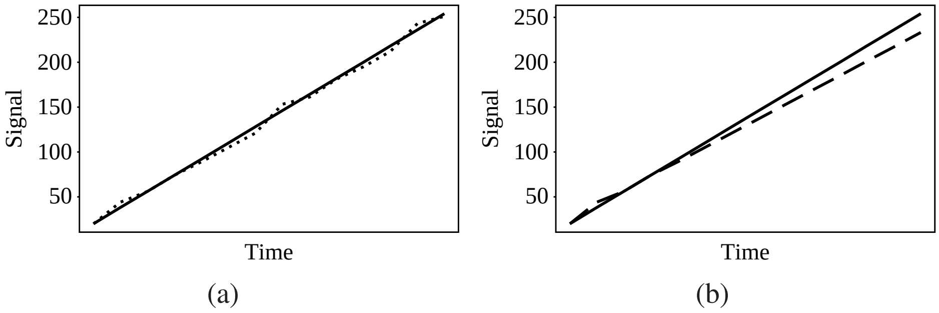


Fig. 6.1: (a) DPCM reconstructed signal (dotted line) tracks the input signal (solid line). (b) DPCM reconstructed signal (dashed line) steers farther and farther from the input signal (solid line).

22. (a) Suppose we use a predictor as follows:

$$\begin{aligned} \hat{f}_n &= \text{trunc}\left(\tfrac{1}{2}(\tilde{f}_{n-1} + \tilde{f}_{n-2})\right) \quad , \\ e_n &= f_n - \hat{f}_n \quad . \end{aligned} \tag{6.2}$$

Also, suppose we adopt the quantizer Eq. (6.20). If the input signal has values as follows:

20 38 56 74 92 110 128 146 164 182 200 218 236 254

then show that the output from a DPCM coder (without entropy coding) is as follows:

20 44 56 74 89 105 121 153 161 181 195 212 243 251.

Fig. 6.1(a) shows how the quantized reconstructed signal tracks the input signal.

(b) Now, suppose by mistake on the coder side we inadvertently use the predictor for *lossless coding*, Eq. (6.14), using original values f_n instead of quantized ones, $\tilde{f}_n$. Show that on the decoder side we end up with reconstructed signal values as follows:

20 44 56 74 89 105 121 137 153 169 185 201 217 233,

so that the error gets progressively worse.

Fig. 6.1(b) shows how this appears: the reconstructed signal gets progressively worse.

Chapter 7

Lossless Compression Algorithms

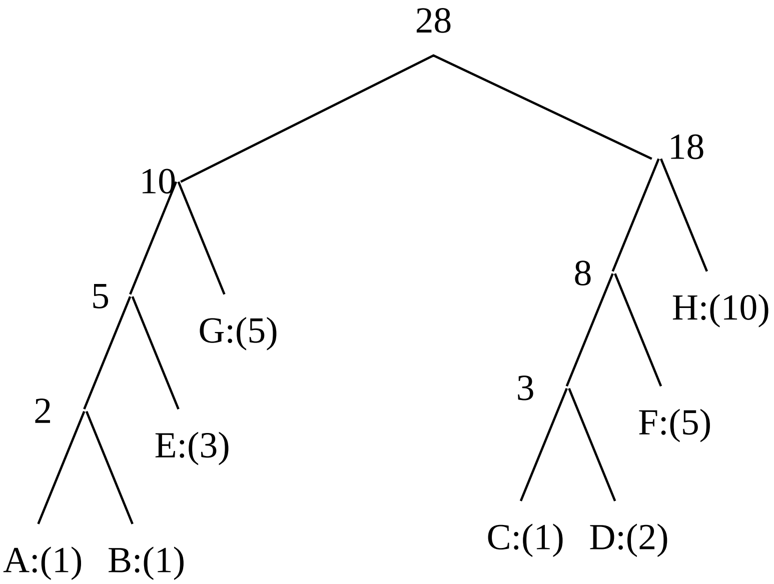
Exercises

1. Calculate the *entropy* of a "checkerboard" image in which half of the pixels are BLACK and half of them are WHITE.

   Answer:
   1

2. Suppose eight characters have a distribution A:(1), B:(1), C:(1), D:(2), E:(3), F:(5), G:(5), H:(10). Draw a Huffman tree for this distribution. (Because the algorithm may group subtrees with equal probability in a different order, your answer is not strictly unique.)

   Answer:



3. (a) What is the entropy η of the image below, where numbers (0, 20, 50, 99) denote the graylevel intensities?

$$\begin{array}{cccccccc}
99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\
20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 50 & 50 & 50 & 50 & 0 & 0 \\
0 & 0 & 50 & 50 & 50 & 50 & 0 & 0 \\
0 & 0 & 50 & 50 & 50 & 50 & 0 & 0 \\
0 & 0 & 50 & 50 & 50 & 50 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

(b) Show step by step how to construct the Huffman tree to encode the above four intensity values in this image. Show the resulting code for each intensity value.

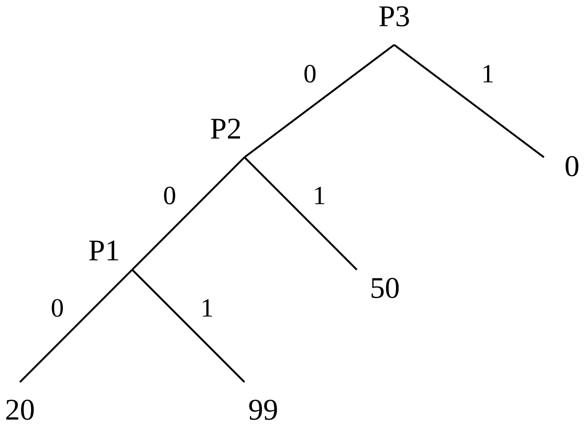
(c) What is the average number of bits needed for each pixel, using your Huffman code? How does it compare to η?

Answer:

(a) $P_{20} = P_{99} = 1/8$, $P_{50} = 1/4$, $P_0 = 1/2$.

$$\eta = 2 \times \frac{1}{8} \log_2 8 + \frac{1}{4} \log_2 4 + \frac{1}{2} \log_2 2 = \frac{3}{4} + \frac{1}{2} + \frac{1}{2} = 1.75$$

(b) Only the final tree is shown below. Resulting code: 0: "1", 50: "01", 20: "000", 99: "001"



(c) Average number of bits = $0.5 \times 1 + 0.25 \times 2 + 2 \times 0.125 \times 3 = 1.75$.

4. Consider an alphabet with two symbols A, B, with probability $P(A) = x$ and $P(B) = 1 - x$.

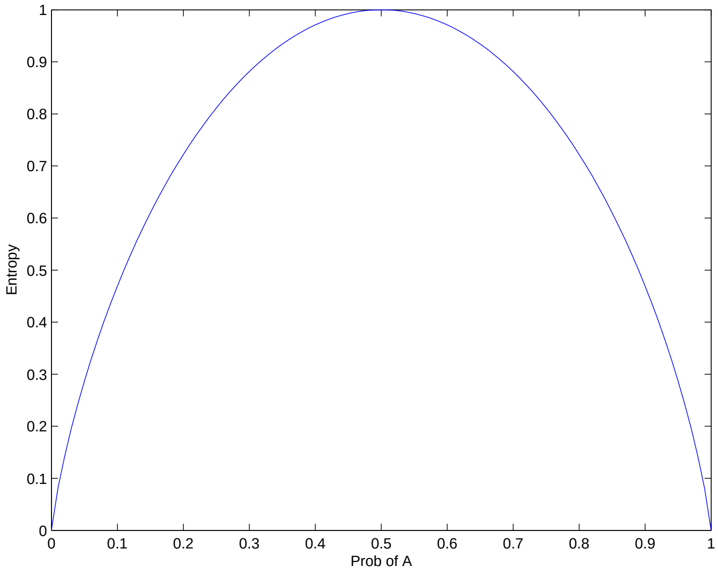
(a) Plot the entropy as a function of x. You might want to use $\log_2 3 = 1.6$ and $\log_2 7 = 2.8$.

(b) Discuss why it must be the case that if the probability of the two symbols is $1/2 + \epsilon$ and $1/2 - \epsilon$, with small ϵ, the entropy is less than the maximum.

(c) Generalize the above result by showing that, for a source generating N symbols, the entropy is maximum when the symbols are all equiprobable.

(d) As a small programming project, write code to verify the conclusions above.

Answer:



5. Extended Huffman Coding assigns one codeword to each group of k symbols. Why is $average(l)$ (the average number of bits for each symbol) still no less than the entropy η as indicated in Eq. (7.7)?

   Answer:
   For extended Huffman coding

$$\eta \leq \bar{l} < \eta + \frac{1}{k}.$$

   — that is, the average number of bits for each symbol is still no less than the entropy η.

6. (a) Suppose we are coding a *binary* source, i.e., the alphabet consists of 0 or 1. For example, a fax is like this.

   Suppose the probability of a 0 is 7/8, and that for a 1 is 1/8;

   What is the entropy? (Note: In case you don't have a calculator, log2(7)= 2.8074)

   Answer:
   -7/8*log2(7/8)-1/8*log2(1/8) = 0.5436 bit

   What is the set of Huffman codes? And what is the average bitrate?

   Answer:
   Huffman code: 0, 1; Avg. code length: 1 bit

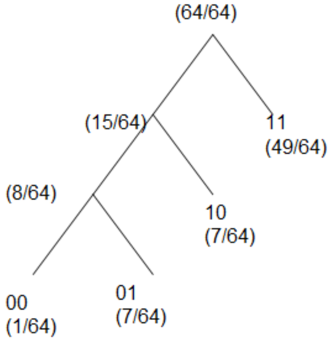
   (b) Now code the problem in terms of Extended Huffman compression, using $n = 2$ and groups of $k = 2$ pairs of symbols. What is the average bitrate now? Show your work (you can just use fractions, not decimal numbers, if you like).

   Answer:
   Using Extended Huffman, whereby more than a single symbol is coded at one time:

Symbol group	**Probability**	**Codeword**	**Bitlength**
00	**1/64**	**000**	**3**
01	**7/64**	**001**	**3**
10	**7/64**	**01**	**2**
11	**29/64**	**1**	**1**

   The probabilities sum to 1.

Code-lengths come from the Huffman tree:



Consequently, the average bitrate per symbol (i.e., per single symbol, so have to multiply by 1/2) is: 87/128 = 0.6797

7. (a) What are the advantages and disadvantages of Arithmetic Coding as compared to Huffman Coding?

   Answer:

   The main advantage of Arithmetic Coding over Huffman Coding is that whereas the minimum code length for a symbol in Huffman Coding is 1, since we create a binary tree with 0 or 1 attached to each branch, in Arithmetic Coding the number of bits per symbol can be fractional.

   (b) Suppose the alphabet is $[A, B, C]$, and the known probability distribution is $P_A = 0.5, P_B = 0.4, P_C = 0.1$. For simplicity, let's also assume that both encoder and decoder know that the length of the messages is always 3, so there is no need for a terminator.

   i. How many bits are needed to encode the message BBB by Huffman coding?
   Answer:
   6 bits.

   ii. How many bits are needed to encode the message BBB by Arithmetic coding?
   Answer:
   The solution is 4 bits.

8. (a) What are the advantages of Adaptive Huffman Coding compared to the original Huffman Coding algorithm?

   (b) Assume that Adaptive Huffman Coding is used to code an information source S with a vocabulary of four letters (a, b, c, d). Before any transmission, the initial coding is a = 00, b = 01, c = 10, d = 11. As in the example illustrated in Figure 7.8, a special symbol NEW will be sent before any letter if it is to be sent the first time.

   Figure 7.1 is the Adaptive Huffman tree after sending letters **aabb**. After that, the additional bitstream received by the decoder for the next few letters is 01010010101.

   i. What are the additional letters received?

   ii. Draw the adaptive Huffman trees after each of the additional letters is received.

   Answer:

   (i) The additional letters received are "b (01) a (01) c (00 10) c (101)".

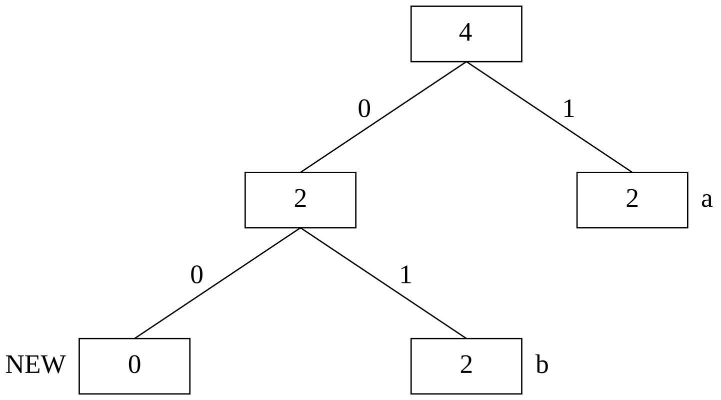
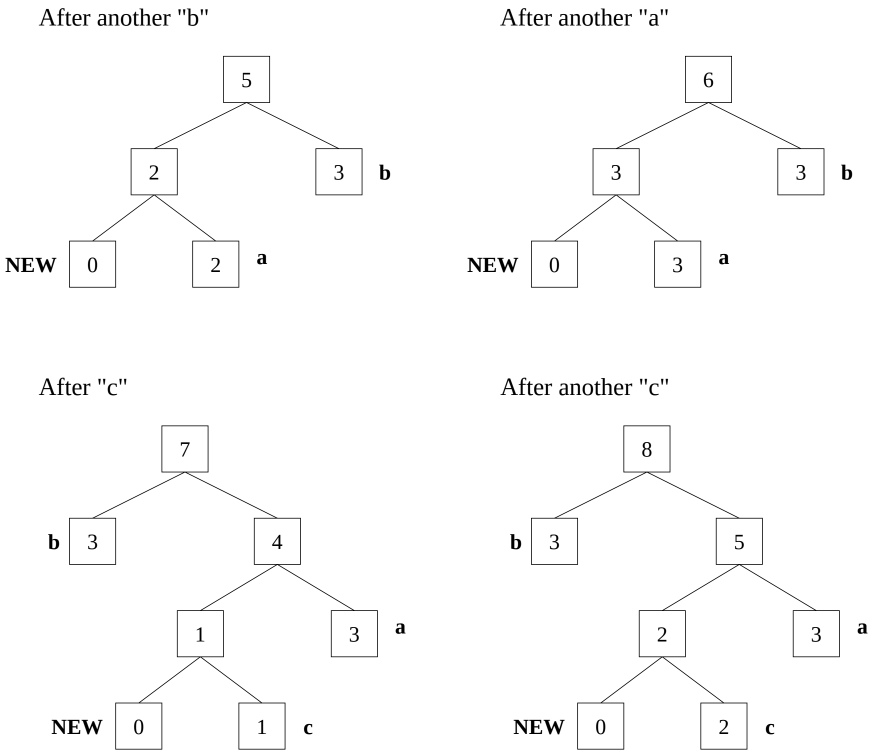


Fig. 7.1: Adaptive Huffman tree.

(ii) The trees are as below.


After another "b"

After another "a"

After "c"

After another "c"

9. Work out the details of Scaling and incremental coding in Arithmetic coding when the probabilities for the three symbols are A: 0.8, B: 0.02, C: 0.18, and the input sequence is ACBA.

10. Work out the details of the encoder and decoder for Adaptive Arithmetic coding when the input symbols are 01111.

11. Compare the rate of adaptation of Adaptive Huffman coding and Adaptive Arithmetic coding. What prevents each method from adapting to quick changes in source statistics?

    Answer:
    Both methods would have a similar rate of adaptation since they both use symbol occurrences as estimates of symbol probability. In the adaptive Huffman case, the symbol occurrences are

used as weights on each node to enforce the sibling property while adaptive arithmetic coding uses them to form cumulative frequency table.

What prevents both algorithms to adapt to quick changes in input statistics is that symbols must occur enough times according to their probablity in order for the Huffman tree and the cumulative frequency table to emerge into a form that reflects the intended probabilities.

12. Consider the dictionary-based LZW compression algorithm. Suppose the alphabet is the set of symbols {`0,1`}. Show the dictionary (symbol sets plus associated codes) and output for LZW compression of the input

```
0 1 1 0 0 1 1
```

Answer:
With input 0 1 1 0 0 1 1, we have

```
                           DICTIONARY
w   k  wk | output | index   symbol
-   -  -- | ------ | -----   ------
NIL 0  0  |        |
0   1  01 | 0      | 2       01
1   1  11 | 1      | 3       11
1   0  10 | 1      | 4       10
0   0  00 | 0      | 5       00
0   1  01 |        |
01  1  011| 2      | 6       011
1         |        |
```

13. Implement Huffman coding, LZW coding, and Arithmetic coding algorithms using your favorite programming language. Generate at least three types of statistically different artificial data sources to test your implementation of these algorithms. Compare and comment on each algorithm's performance in terms of compression ratio for each type of data source.

    Optionally, implement Adaptive Huffman and Adaptive Arithmetic coding algorithms.

Chapter 8

Lossy Compression Algorithms

Exercises

1. Assume we have an unbounded source we wish to quantize using an M-bit midtread uniform quantizer. Derive an expression for the total distortion if the step size is 1.

   Answer:
   The total distortion can be divided into two components: the granular distortion and overload distortion. Let $k = \frac{2^M}{2}$. Since we have an M bit midtread quantizer, the number of reconstruction levels are $k-1$. Since two reconstruction values are allocated for the overload regions, there are $k-3$ reconstruction levels in the granular region. Therefore, we have the following expression for the total distortion.

$$
\begin{aligned}
D &= D_g + D_o \\
&= \left(2 \sum_{i=1}^{\frac{k}{2}-2} \int_{i-0.5}^{i+0.5} (x-i)^2 f_X(x)dx + \int_{-0.5}^{0.5} x^2 f_X(x)dx \right) + \\
&\quad \left(2 \int_{\frac{k}{2}-2}^{\infty} (x - (\frac{k}{2} - 2 + 0.5))^2 f_X(x)dx \right)
\end{aligned}
$$

2. Suppose the domain of a uniform quantizer is $[-b_M, b_M]$. We define the loading fraction as

$$\gamma = \frac{b_M}{\sigma}$$

   where σ is the standard deviation of the source. Write a simple program to quantize a Gaussian distributed source having zero mean and unit variance using a 4-bit uniform quantizer. Plot the SNR against the loading fraction and estimate the optimal step size that incurs the least amount of distortion from the graph.

   Answer:

   The plot should look something like Fig. 8.28.

   If there are M reconstruction levels, then the optimal step size is $\frac{2b^*_M}{M}$, where b^*_M is the value of b_M at which the SNR is maximum on the graph.

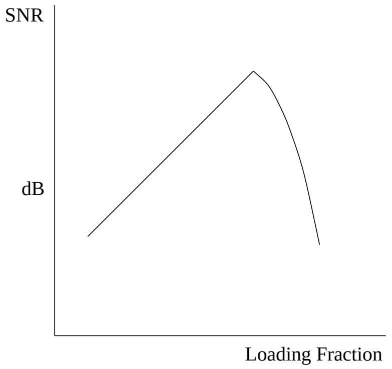


Fig. 8.28: SNR VS Loading Fraction

3. * Suppose the input source is Gaussian-distributed with zero mean and unit variance — that is, the probability density function is defined as

$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{8.66}$$

We wish to find a four-level Lloyd–Max quantizer. Let $\mathbf{y}_i = [y_i^0, \ldots, y_i^3]$ and $\mathbf{b}_i = [b_i^0, \ldots, b_i^3]$. The initial reconstruction levels are set to $\mathbf{y}_0 = [-2, -1, 1, 2]$. This source is unbounded, so the outer two boundaries are $+\infty$ and $-\infty$.

Follow the Lloyd–Max algorithm in this chapter: the other boundary values are calculated as the midpoints of the reconstruction values. We now have $\mathbf{b}_0 = [-\infty, -1.5, 0, 1.5, \infty]$. Continue one more iteration for $i = 1$, using Eq. (8.13) and find y_0^1, y_1^1, y_2^1, y_3^1, using numerical integration. Also calculate the squared error of the difference between $\mathbf{y}_1$ and $\mathbf{y}_0$.

Iteration is repeated until the squared error between successive estimates of the reconstruction levels is below some predefined threshold ϵ. Write a small program to implement the Lloyd–Max quantizer described above.

Answer:

The reconstruction values at $i = 1$ is calculated using Equation (8.13). Using numerical integration, we have

$$y_0^1 = \frac{\int_{-\infty}^{-1.5} \frac{x}{\sqrt{2\pi}} e^{-\frac{x}{2}}\, dx}{\int_{-\infty}^{-1.5} \frac{1}{\sqrt{2\pi}} e^{-\frac{x}{2}}\, dx} = -1.94, \quad y_1^1 = \frac{\int_{-1.5}^{0} \frac{x}{\sqrt{2\pi}} e^{-\frac{x}{2}}\, dx}{\int_{-1.5}^{0} \frac{1}{\sqrt{2\pi}} e^{-\frac{x}{2}}\, dx} = -0.62,$$
$$y_2^1 = \frac{\int_{0}^{1.5} \frac{x}{\sqrt{2\pi}} e^{-\frac{x}{2}}\, dx}{\int_{0}^{1.5} \frac{1}{\sqrt{2\pi}} e^{-\frac{x}{2}}\, dx} = 0.62, \qquad y_3^1 = \frac{\int_{1.5}^{\infty} \frac{x}{\sqrt{2\pi}} e^{-\frac{x}{2}}\, dx}{\int_{1.5}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x}{2}}\, dx} = 1.94.$$

The square error of the difference between $\mathbf{y}_1$ and $\mathbf{y}_0$ is computed as

$$(-1.92 + 2)^2 + (-0.62 + 1)^2 + (0.62 - 1)^2 + (1.92 - 1)^2 = 0.296$$

This process is repeated until the square error between successive estimates of the reconstruction levels are below some predefined threshold ϵ.

4. If the block size for a 2D DCT transform is 8 × 8, and we use only the DC components to create a thumbnail image, what fraction of the original pixels would we be using?

Answer:
1/64, because each 8×8 block only has one DC.

5. When the blocksize is 8, the definition of the DCT is given in Eq. (8.17).

   (a) If an 8×8 grayscale image is in the range $0 \mathinner{..} 255$, what is the largest value a DCT coefficient could be, and for what input image? (Also, state *all* the DCT coefficient values for that image.)

   (b) If we first subtract the value 128 from the whole image and then carry out the DCT, what is the exact effect on the DCT value $F[2,3]$?

   (c) Why would we carry out that subtraction? Does the subtraction affect the number of bits we need to code the image?

   (d) Would it be possible to invert that subtraction, in the IDCT? If so, how?

   Answer:

   (a) When the image is all WHITE, i.e., all pixels have $I = 255$. The largest coefficient is the DC value which is $8 \times 255 = 2,040$. All others (AC values) are zero.

   (b) There is no effect on $F[2,3]$. In fact, no effect on any AC values.

   (c) The idea here is to turn it into a zero mean image, so we do not waste any bits in coding the mean value. (Think of an 8×8 block with intensity values ranging from 120 to 135.)

   (d) After decoding, simply add 128 back to all pixel values.

6. Write a simple program or refer to the sample DCT program `dct_1D.c` in the book's web site to verify the results in Example 8.2 of the 1D DCT example in this chapter.

7. Write a program to verify that the DCT-matrix $\mathbf{T_8}$ as defined in Eqs. 8.29 and 8.30 is an Orthogonal Matrix, i.e., all its rows and columns are orthogonal unit vectors (orthonormal vectors).

8. Write a program to verify that the 2D DCT and IDCT matrix implementations as defined in Eqs. 8.27 and 8.31 are lossless, i.e., they can transform any 8×8 values $f(i,j)$ to $F(u,v)$ and back to $f(i.j)$. (Here, we are not concerned with possible/tiny floating point calculation errors.)

9. We could use a similar DCT scheme for *video streams*, by using a 3D version of DCT.

   Suppose one color-component of a video has pixels f_{ijk} at position (i,j) and time k. How could we define its 3D DCT transform?

   Answer:

$$\begin{aligned} F[u,v,w] &= \frac{1}{4}\sqrt{\frac{2}{N}} C(u)C(v)C(w) \sum_{i=0}^{7}\sum_{j=0}^{7}\sum_{k=0}^{N-1} \cos\frac{(2i+1)u\pi}{16} \\ & \quad \cdot \cos\frac{(2j+1)v\pi}{16} \cdot \cos\frac{(2k+1)w\pi}{2N} \cdot f[i,j,k] \end{aligned}$$

   Must decide on the number N of frames.

10. Suppose a uniformly colored sphere is illuminated and has shading varying smoothly across its surface, as in Fig. 8.29.

    (a) What would you expect the DCT coefficients for its image to look like?

    (b) What would be the effect on the DCT coefficients of having a checkerboard of colors on the surface of the sphere?

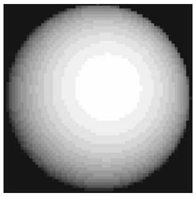


Fig. 8.29: Sphere shaded by a light.

(c) For the uniformly colored sphere again, describe the DCT values for a block that straddles the top edge of the sphere, where it meets the black background.

(d) Describe the DCT values for a block that straddles the left edge of the sphere.

Answer:

(a) For the top edge of the sphere, there are DCT values along the left column of each 8×8 DCT block and near zero elsewhere. This is because the cosine functions with $v = 0$ are higher and higher frequencies in the y-direction, and no change in the x-direction; so there is a contribution to these cosines — whereas the cosines with $u = 0$ are sinusoids changing in the x-direction and not in the y-direction, but there is no corresponding change in the x-direction at the top of the sphere, so there result DCTs with values near zero.

For the left (or right) edge of the sphere, there are DCT values along the top row of each 8×8 and near zero elsewhere.

(b) Still lots of low-frequency; but also more higher frequencies because of the sharp edges of the beach ball colors.

(c) For the block that straddle the top edge, as in (a) there are DCT values along the left column of each 8×8 block and near zero elsewhere, although some coefficients in the column, e.g., $F(1, 0)$, will now have much larger (absolute) value in response to the change from black to grey.

(d) For the block that straddle the left edge, this change happens in the top row of each 8×8 block.

11. The Haar wavelet has a scaling function which is defined as follows:

$$\phi(t) \;=\; \begin{cases} 1 & 0 \le t \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{8.67}$$

and its scaling vector is $h_0[0] = h_0[1] = 1/\sqrt{2}$.

(a) Draw the scaling function, and then verify that its dilated translates $\phi(2t)$ and $\phi(2t - 1)$ satisfy the dilation equation (8.56). Draw the combination of these functions that make up the full function $\phi(t)$.

(b) Derive the wavelet vector $h_1[0]$, $h_1[1]$ from Eq. (8.59) and then derive and draw the Haar wavelet function $\psi(t)$ from Eq. (8.57).

Answer:

(a) $\phi(t) = \sqrt{2}\,[\frac{1}{\sqrt{2}}\,\phi(2t) + \frac{1}{\sqrt{2}}\,\phi(2t-1)] = \phi(2t) + \phi(2t-1)$, **so that two half-steps make up the full step.**

The two parts are:

$$\phi(2t) \;=\; \begin{cases} 1 & 0 \leq t < 0.5 \\ 0 & \textbf{otherwise} \end{cases}$$

and

$$\phi(2t-1) \;=\; \begin{cases} 1 & 0.5 \leq t < 1 \\ 0 & \textbf{otherwise} \end{cases}$$

(b) $h_1[0] = h_0[1] = 1/\sqrt{2},\, h_1[1] = -h_0[0] = -1/\sqrt{2},$ **so**
$\psi(t) = \sqrt{2}\,[h_1[0]\;\phi(2t) + h_1[1]\;\phi(2t-1)] = \phi(2t) - \phi(2t-1).$

$$\psi(t) \;=\; \begin{cases} 1 & 0 \leq t < 0.5 \\ -1 & 0.5 \leq t < 1 \\ 0 & \textbf{otherwise} \end{cases}$$

12. Suppose the mother wavelet $\psi(t)$ has vanishing moments M_p up to and including M_n. Expand $f(t)$ in a Taylor series around $t = 0$, up to the nth derivative of f [i.e., up to leftover error of order $O(n+1)$]. Evaluate the summation of integrals produced by substituting the Taylor series into (8.52) and show that the result is of order $O(s^{n+2})$.

    Answer:

    $\mathcal{W}(f,s,u) = \frac{1}{\sqrt{s}} \int f(t)\psi(\frac{t-u}{s})dt$
    $= \frac{1}{\sqrt{s}} \sum_{p=0}^{n} f^{(p)}(0) \int (t^p/p!)\psi((t-u)/s)dt$
    and let $u == 0$**.**

13. The program `wavelet_compression.c` on this book's web site is in fact simple to implement as a MATLAB function (or similar fourth-generation language). The advantage in doing so is that the `imread` function can input image formats of a great many types, and `imwrite` can output as desired. Using the given program as a template, construct a MATLAB program for wavelet-based image reduction, with perhaps the number of wavelet levels being a function parameter.

14. It is interesting to find the Fourier transform of functions, and this is easy if you have available a symbolic manipulation system such as MAPLE. In that language, you can just invoke the `fourier` function and view the answer directly! As an example, try the following code fragment:

    ```
    with(`inttrans`); f := 1; F := fourier(f,t,w);
    ```

    The answer should be $2\pi\delta(w)$. Let's try a Gaussian:

    ```
    f := exp(-t^2);
    F := fourier(f,t,w);
    ```

    Now the answer should be $\sqrt{\pi}e^{(-w^2/4)}$: the Fourier transform of a Gaussian is simply another Gaussian.

15. Suppose we define the wavelet function

$$\psi(t) \;=\; exp(-t^{1/4})\sin(t^4)\;, \quad t \geq 0 \tag{8.68}$$

This function oscillates about the value 0. Use a plotting package to convince yourself that the function has a zero moment M_p for any value of p.

Answer:
To see that function $\psi(t) = e^{(-t^{(1/4)})}\sin(t^4)$ has all zero moments, use Matlab, or Maple:

```
% In Matlab: moments:
t = 0:0.001:1000;
t = t';
psi = exp(-t.(̂1/4)) .* sin(t.4̂);
plot(t,psi,'.');
xlabel('t');
ylabel('ψ(t)');
print -depsc 'zeromoments.eps'; % as in Fig. 8.30.
mom3 = t.3̂.*psi;
0.001* sum(mom3)/max(mom3) % approx zero
mom5 = t5̂*psi;
0.001* sum(mom5)/max(mom5) % approx zero
```

And to see more theoretically,

```
# In Maple: moments:
f := exp(-t(̂1/4)) * sin(t4̂);
#Mp := int(f*tp̂, t=0..infinity);
# Use numerical integration:
M1 := evalf(int(f*t, t=0..1)); # 0.05954452727, and max is 0.3096 in t=0..1
M2 := evalf(int(f*t2̂, t=0..1)); # 0.05040557254
M3 := evalf(int(f*t3̂, t=0..1)); # 0.04365545560
M7 := evalf(int(f*t7̂, t=0..1)); # 0.02829949838
```

16. Implement both a DCT-based and a wavelet-based image coder. Design your user interface so that the compressed results from both coders can be seen side by side for visual comparison. The PSNR for each coded image should also be shown, for quantitative comparisons.

Include a slider bar that controls the target bitrate for both coders. As you change the target bitrate, each coder should compress the input image in real time and show the compressed results immediately on your user interface.

Discuss both qualitative and quantitative compression results observed from your program at target bitrates of 4 bpp, 1 bpp, and 0.25 bpp.

Answer:

See example in the Demo part of the web site.

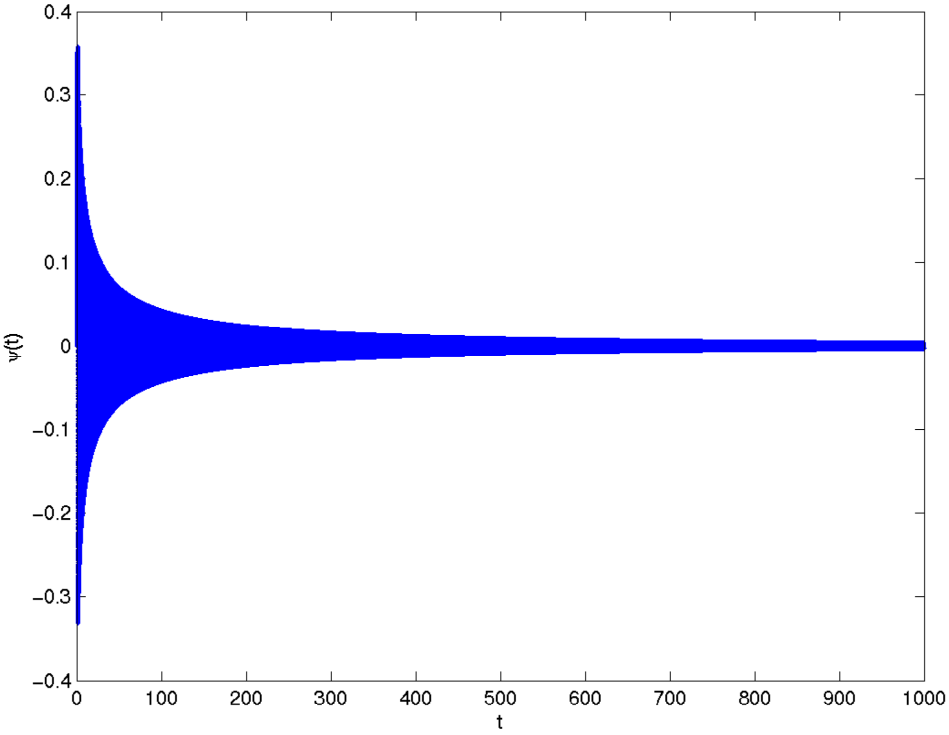


Fig. 8.30: Highly oscillating values function with all zero moments.

Chapter 9

Image Compression Standards

Exercises

1. You are given a computer cartoon picture and a photograph. If you have a choice of using either JPEG compression or GIF, which compression would you apply for these two images? Justify your answer.

   Answer:

   GIF will generally be better, compared to JPEG, because JPEG firstly keeps 24-bit color, which is not needed for a cartoon, which has only a few numbers, and as well the LZW compression will do a very good job in terms of compression ratio, since only a few values exist and their runs will build up in the dictionary. On the other hand, JPEG will treat the image as a complex image no matter if it is or not.

2. Suppose we view a decompressed 512×512 JPEG image but use only the *color* part of the stored image information, not the luminance part, to decompress. What does the 512×512 color image look like? Assume JPEG is compressed using a 4:2:0 scheme.

   Answer:

   Without all components, we cannot restore the color image. Assuming that we are only recovering the monochrome part, then first, each pixel is an enlarged (duplicated) version of a *subsampled* 2×2 **block, so the image is "pixellated" looking — blocky. Second, although the luminance and chrominance images are often correlated, there is no guarantee that they will capture the same shape and texture information. Depending on the image content, the image displayed may be highly distorted in terms of shape and texture.**

3. An X-ray photo is usually a graylevel image. It often has low contrast and low graylevel intensities, i.e., all intensity values are in the range of $[a, b]$, where a and b are positive integers, much less than the maximum intensity value 255 if it is an 8-bit image. In order to enhance the appearance of this image, a simple "stretch" operation can be used to convert all original intensity values f_0 to f:

   $$f = \frac{255}{b-a} \cdot (f_0 - a).$$

   For simplicity, assuming $f_0(i, j)$ and $f(i, j)$ are 8×8 images:

   (a) If the DC value for the original image f_0 is m, what is the DC value for the stretched image f?

   (b) If one of the AC values $F_0(2, 1)$ for the original image f_0 is n, what is the $F(2, 1)$ value for the stretched image f?

Answer:

DC:

$$\frac{255}{b-a} \cdot m - 8 \cdot \frac{255}{b-a} \cdot a$$

$F(2,1)$**:**

$$\frac{255}{b-a} \cdot n$$

4. (a) JPEG uses the Discrete Cosine Transform (DCT) for image compression.
   i. What is the value of F(0, 0) if the image $f(i,j)$ is as below?
   ii. Which AC coefficient $|F(u,v)|$ is the largest for this $f(i,j)$? Why? Is this $F(u,v)$ positive or negative? Why?

$$\begin{matrix}
20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 \\
20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 \\
80 & 80 & 80 & 80 & 80 & 80 & 80 & 80 \\
80 & 80 & 80 & 80 & 80 & 80 & 80 & 80 \\
140 & 140 & 140 & 140 & 140 & 140 & 140 & 140 \\
140 & 140 & 140 & 140 & 140 & 140 & 140 & 140 \\
200 & 200 & 200 & 200 & 200 & 200 & 200 & 200 \\
200 & 200 & 200 & 200 & 200 & 200 & 200 & 200
\end{matrix}$$

(b) Show in detail how a three-level hierarchical JPEG will encode the image above, assuming that
   i. The encoder and decoder at all three levels use Lossless JPEG.
   ii. *Reduction* simply averages each 2×2 block into a single pixel value.
   iii. *Expansion* duplicates the single pixel value four times.

Answer:

(a) i. 8 times average-intensity $= 8 \times 110 = 880$**.**

ii. $|F(1,0)|$ **is the largest, because the intensity value change is similar to a half cosine cycle vertically within the** 8×8 **block.** $F(1,0)$ **is negative, because the phase of the change is off by 180 degrees. (Or simply put, it is opposite.)**

(b) Step by step results:

X_2**:**

$$\begin{matrix}
\mathbf{20} & \mathbf{20} & \mathbf{20} & \mathbf{20} \\
\mathbf{80} & \mathbf{80} & \mathbf{80} & \mathbf{80} \\
\mathbf{140} & \mathbf{140} & \mathbf{140} & \mathbf{140} \\
\mathbf{200} & \mathbf{200} & \mathbf{200} & \mathbf{200}
\end{matrix}$$

X_4**:**

$$\begin{matrix}
\mathbf{50} & \mathbf{50} \\
\mathbf{170} & \mathbf{170}
\end{matrix}$$

$E(X_4)$**:**

$$\begin{matrix}
\mathbf{50} & \mathbf{50} & \mathbf{50} & \mathbf{50} \\
\mathbf{50} & \mathbf{50} & \mathbf{50} & \mathbf{50} \\
\mathbf{170} & \mathbf{170} & \mathbf{170} & \mathbf{170} \\
\mathbf{170} & \mathbf{170} & \mathbf{170} & \mathbf{170}
\end{matrix}$$

D_2**:**

$$\begin{matrix}
\mathbf{-30} & \mathbf{-30} & \mathbf{-30} & \mathbf{-30} \\
\mathbf{30} & \mathbf{30} & \mathbf{30} & \mathbf{30} \\
\mathbf{-30} & \mathbf{-30} & \mathbf{-30} & \mathbf{-30} \\
\mathbf{30} & \mathbf{30} & \mathbf{30} & \mathbf{30}
\end{matrix}$$

$E(X_2) = X_1$
(same as X_1, not shown)

D_2**:**

$$\begin{array}{cccccccc}
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}
\end{array}$$

Assuming P1 mode of Lossless JPEG (i.e., take the immediate preceding pixel as the predicted value), then the codewords generated are:

X_4**: 50 0 120 0**

D_2**: -30 0 0 0 60 0 0 0 -60 0 0 0 60 0 0 0**

D_1**: 0 0 0 $\ldots$ 0 0**

5. In JPEG, the Discrete Cosine Ttransform is applied to 8×8 blocks in an image. For now, let's call it DCT-8. Generally, we can define a DCT-N to be applied to $N \times N$ blocks in an image. DCT-N is defined as:

$$\begin{aligned}
F_N(u,v) &= \frac{2C(u)C(v)}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \cos\frac{(2i+1)u\pi}{2N} \cos\frac{(2j+1)v\pi}{2N} f(i,j) \\
C(\xi) &= \begin{cases} \frac{\sqrt{2}}{2} & \text{for } \xi = 0 \\ 1 & \text{otherwise} \end{cases}
\end{aligned}$$

Given $f(i,j)$ as below, show your work for deriving all pixel values of $F_2(u,v)$. (That is, show the result of applying DCT-2 to the image below.)

$$\begin{array}{cccccccc}
100 & -100 & 100 & -100 & 100 & -100 & 100 & -100 \\
100 & -100 & 100 & -100 & 100 & -100 & 100 & -100 \\
100 & -100 & 100 & -100 & 100 & -100 & 100 & -100 \\
100 & -100 & 100 & -100 & 100 & -100 & 100 & -100 \\
100 & -100 & 100 & -100 & 100 & -100 & 100 & -100 \\
100 & -100 & 100 & -100 & 100 & -100 & 100 & -100 \\
100 & -100 & 100 & -100 & 100 & -100 & 100 & -100 \\
100 & -100 & 100 & -100 & 100 & -100 & 100 & -100
\end{array}$$

Answer:

Divide the image into 2 by 2 blocks. We only need to work out the four coefficients for $F_2(u,v)$, then they'll repeat.

$F_2(0,0) = 0$**, because average intensity is zero.**

$F_2(1,0) = 0$**, because no change vertically.**

$F_2(0,1) = \frac{\sqrt{2}}{2}\left[\cos\frac{\pi}{4}\cdot 100 + \cos\frac{3\pi}{4}\cdot(-100) + \cos\frac{\pi}{4}\cdot 100 + \cos\frac{3\pi}{4}\cdot(-100)\right] = \frac{\sqrt{2}}{2}\cdot\frac{\sqrt{2}}{2}\cdot 4\cdot 100 = 200$.

$F_2(1,1) = 0$**, because the signal and the basis function are orthogonal. (Students may check by calculating numbers.)**

$F_2(u,v)$**:**

0	**200**	**0**	**200**	**0**	**200**	**0**	**200**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**200**	**0**	**200**	**0**	**200**	**0**	**200**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**200**	**0**	**200**	**0**	**200**	**0**	**200**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**200**	**0**	**200**	**0**	**200**	**0**	**200**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**

6. According to the DCT-N definition above, $F_N(1)$ and $F_N(N-1)$ are the AC coefficients representing the lowest and highest spatial frequencies, respectively.

   (a) It is known that $F_{16}(1)$ and $F_8(1)$ *do not* capture the same (lowest) frequency response in image filtering. Explain why.

   (b) Do $F_{16}(15)$ and $F_8(7)$ capture the same (highest) frequency response?

   Answer:

   First, we need a 1D DCT-N definition:

$$F_N(u) = \sqrt{\frac{2}{N}}\, C(u) \sum_{i=0}^{N-1} \cos\frac{(2i+1)\cdot u\pi}{2N}\cdot f(i)$$

   (a) The basis function for $F_8(1)$ **is** $\cos\frac{(2i+1)\cdot\pi}{16}$**, for** $F_{16}(1)$ **it is** $\cos\frac{(2i+1)\cdot\pi}{32}$**. Hence,** $F_8(1)$ **corresponds to changes of half a cosine cycle within a distance of 16 pixels; whereas** $F_{16}(1)$ **corresponds to half a cosine cycle within a distance of 32 pixels — a frequency twice as low.**

   (b) Yes, $F_{16}(15)$ **and** $F_8(7)$ **capture the same (highest) frequency response, because they both capture the highest possible frequency — the 1D signal oscillates between white and black for every pixel, which corresponds to half a cosine cycle within a distance of 1 pixel.**

7. (a) How many principal modes does JPEG have? What are their names?

   (b) In the hierarchical model, explain briefly why we must include an encode/decode cycle on the coder side before transmitting difference images to the decode side.

(c) What are the two methods used to decode only part of the information in a JPEG file, so that the image can be coarsely displayed quickly and iteratively increased in quality?

Answer:

(a) 4: baseline sequential, progressive, hierarchical, lossless.

(b) Because the encoder must only use quantized values that the decoder (receiver) has to operate on.

(c) 1. Spectral selection – decode using few and then progressively more (and higher-frequency) coefficients of the DCT matrix for each block.
2. Successive approximation – decode using only the MSB in the DCT representation first, and then progressively add more bits to decode.

8. Could we make use of wavelet-based compression in ordinary JPEG? How?

Answer:

Use a series of combinations of Haar-type scaled and shifted wavelets instead of DCT curves which look much like the actual DCT basis functions.

9. We decide to create a new image-compression standard based on JPEG, for use with images that will be viewed by an alien species. What part of the JPEG workflow would we likely have to change?

Answer:

The quantization tables, which are based on the human visual system's perceptual importance weighting for different spatial frequencies.

10. Unlike EZW, EBCOT does not explicitly take advantage of the spatial relationships of wavelet coefficients. Instead, it uses the PCRD optimization approach. Discuss the rationale behind this approach.

Answer:

By not explicitly exploiting spatial relationships in subbands, the Post Compression Rate Distortion (PCRD) optimization approach is able to offer several advantages. First, the PCRD approach allows subbands to be divided into blocks that are compressed independently. The rate distortion optimization is then performed after all code blocks have been compressed. This gives the encoder a more global view of the code block samples and thus providing better rate distortion tradeoffs.

Second, the layered coding plus PCRD allows the final bitstream to be both resolution and SNR scalable, whereas algorithms such as EZW can only construct SNR scalable bitstreams.

11. Is the JPEG2000 bitstream SNR scalable? If so, explain how it is achieved using the EBCOT algorithm.

Answer:

Depending on the number of quality layers, the JPEG2000 bitstream can be either SNR scalable or not. It is not SNR scalable if there is only one single quality layer. With more than one quality layers, it is SNR scalable. SNR scalability in JPEG2000 is achieved using a combination of layered coding and post compression rate distortion (PCRD) optimization. At a particular quality layers, the PCRD algorithm finds the set of truncation points associated with each code block that contributes to this quality layers. Each additional layer in the bitstream refines the previous layer with the specified truncation points. Thus, the final bitstream is SNR scalable.

12. Implement transform coding, quantization, and hierarchical coding for the encoder and decoder of a three-level Hierarchical JPEG. Your code should include a (minimal) graphical user interface for the purpose of demonstrating your results. You do not need to implement the entropy (lossless) coding part; optionally, you may include any publicly available code for it.

    Answer:

    See some examples in the Demo part of the web site.

Chapter 10

Basic Video Compression Techniques

Exercises

1. Describe how H.261 deals with *temporal* and *spatial* redundancies in video.

2. An H.261 video has the three color channels Y, C_r, C_b. Should $\mathbf{MV}$s be computed for each channel and then transmitted? Justify your answer. If not, which channel should be used for motion compensation?

   Answer:
   No. MVs are usually generated from the Y-frames. In almost all cases, the luminance (Y) channel carries sufficient motion information.

3. Thinking about my large collection of `JPEG` images (of my family taken in various locales), I decide to unify them and make them more accessible by simply combining them into a big H.261-compressed file. My reasoning is that I can simply use a viewer to step through the file, making a cohesive whole out of my collection. Comment on the utility of this idea, in terms of the compression ratio achievable for the set of images.

   Answer:
   This will not achieve a good compression, since no temporal redundancy is available. And it may be worse, since extra header information is required.

4. In block-based video coding, what takes more effort: compression or decompression? Briefly explain why.

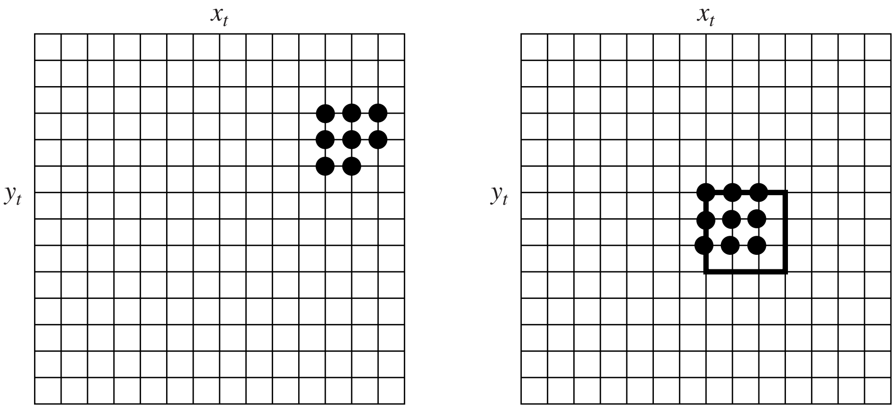
   Answer:
   Compression. The encoder needs to do Motion Compensation (generate the motion vectors) which is time-consuming.

5. Work out the following problem of 2D Logarithmic Search for motion vectors in detail (see Fig. 10.14).

   The target (current) frame is a P-frame. The size of macroblocks is 4×4. The motion vector is $\mathbf{MV}(\Delta x, \Delta y)$, in which $\Delta x \in [-p, p]$, $\Delta y \in [-p, p]$. In this question, assume $p \equiv 5$.

   The macroblock in question (darkened) in the frame has its upper left corner at (x_t, y_t). It contains 9 dark pixels, each with intensity value 10; the other 7 pixels are part of the background, which has a uniform intensity value of 100. The reference (previous) frame has 8 dark pixels.

   (a) What is the best Δx, Δy, and Mean Absolute Error (MAE) for this macroblock?



Reference frame Target frame

● Pixel with intensity value 10

Other background (unmarked) pixels all have intensity value 100

Fig. 10.14: 2D Logarithmic search for motion vectors.

(b) Show step by step how the 2D Logarithmic Search is performed, include the locations and passes of the search and all intermediate Δx, Δy, and MAEs.

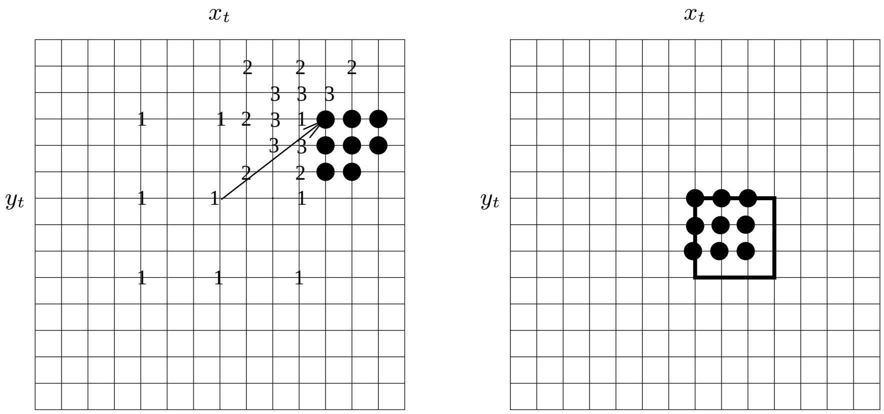
Answer:

(a) $\Delta x = 4$, $\Delta y = 3$. **MAE =** $\frac{100-10}{16} = 5.625$.

(b) Pass 1: $\lceil p/2 \rceil = 3$, $\Delta x = 3$, $\Delta y = 3$, **MAE =** $\frac{3\cdot|100-10|+2\cdot|10-100|}{16} = \frac{450}{16} = 28.125$.

Pass 2: $\lceil p/4 \rceil = 2$, $\Delta x = 3$**,** $\Delta y = 3$, **MAE =** $\frac{3\cdot|100-10|+2\cdot|10-100|}{16} = \frac{450}{16} = 28.125$.

Pass 3: $\lceil p/8 \rceil = 1$, $\Delta x = 4$, $\Delta y = 3$, **MAE =** $\frac{100-10}{16} = 5.625$.



Reference frame Target frame

● − pixel with intensity value 10

Other background (unmarked) pixels all have intensity value 100

Answer to Q5: 2D Logarithmic search for motion vectors.

6. The logarithmic **MV** search method is suboptimal, in that it relies on continuity in the residual frame.

(a) Explain why that assumption is necessary, and offer a justification for it.

(b) Give an example where this assumption fails.

(c) Does the hierarchical search method suffer from suboptimality too?

Answer:

(a) The continuity assumption is based on the observation that visual content of the macroblock and the surrounding macroblocks usually change continuously within a short distance, e.g., a couple dozen pixels away, especially within a short time, e.g., 33 milliseconds for 30 fps. This often turns into monotonicity, i.e., the neighborhood that yielded minimal error in the previous pass will indeed yield another (global) minimum in the current pass and beyond.

(b) The above assumption is too strong to be true all the time. In a highly textured frame, it can easily be the case that the (global) minimum actually is in the neighborhood that did not yield minimal error in the previous pass.

(c) Yes, but less so. The textured frame will again be a possible example. When image resolution changes, certain texture patterns change (or disappear). Hence, the area that yielded minimal error at a lower resolution may not necessarily be a good area to search at the current resolution.

7. A video sequence is given to be encoded using H.263 in PB-mode, having a frame size of 4CIF, frame rate of 30 fps, and video length of 90 minutes. The following is known about the compression parameters: on average, two I-frames are encoded per second. The video at the required quality has an I-frame average compression ratio of 10:1, an average P-frame compression ratio twice as good as I-frame, and an average B-frame compression ratio twice as good as P-frame. Assuming the compression parameters include all necessary headers, calculate the encoded video size.

Answer:

Because of the PB-mode, we can assume P- and B-frames always come in pair. Hence, out of 30 frames per second, we have 2 I-frames, 14 P-frames, and 14 B-frames.

4CIF has a resolution of 704×576 **for luminance and** 352×288 **for chrominance images. Assuming 8-bit images, each uncompressed frame has**

$$704 \times 576 + 2(352 \times 288) = 608,256\ bytes \approx 600KB.$$

Consider average compression ratios: I-frame 1/10, P-frame 1/20, B-frame 1/40. For each second, the compressed video has

$$2 \times \frac{1}{10} \times 600 + 14 \times \frac{1}{20} \times 600 + 14 \times \frac{1}{40} \times 600 \approx 750KB.$$

The video size is hence

$$750KB \times 60 \times 90 \approx 4.05GB.$$

8. Assuming a search window of size $2p + 1$, what is the complexity of motion estimation for a QCIF video in the advanced prediction mode of H.263, using

(a) The brute-force (sequential search) method?

(b) The 2D logarithmic method?

(c) The hierarchical method?

Answer:
It is similar to H.261 except the macroblock size N **is 8 which doesn't affect the complexity. The QCIF luminance image size is** $C \times R = 176 \times 144$**. If frame rate is** F**, then the complexity for each method is as below.**

Sequential search method:

$(2p+1)^2 \cdot N^2 \cdot 3 \cdot \frac{C \cdot R}{N \cdot N} \cdot F,$ **i.e.,** $O(p^2 \cdot CRF)$.

2D logarithmic method:

$(8 \cdot (\lceil \log_2 p \rceil + 1) + 1) \cdot N^2 \cdot 3 \cdot \frac{C \times R}{N \cdot N} \cdot F,$ **i.e.,** $O(\mathbf{log} p \cdot CRF)$.

Hierarchical Search:

$\left[\left(2 \left\lceil \frac{p}{4} \right\rceil + 1\right)^2 \left(\frac{N}{4}\right)^2 + 9 \left(\frac{N}{2}\right)^2 + 9N^2 \right] \times 3 \times \frac{C \times R}{N \cdot N} \times F$

$\left[\left(2 \left\lceil \frac{p}{4} \right\rceil + 1\right)^2 \left(\frac{1}{4}\right)^2 + 9 \left(\frac{1}{2}\right)^2 + 9 \right] \times 3 \times CRF.$ **For a relatively small** p **(e.g.,** $p = 15$**), this cost is very low.**

9. Discuss how the advanced prediction mode in H.263 achieves better compression.

   Answer:
   As discussed in the text, the macroblock size is reduced from 16×16 **to** 8×8 **and four MVs are generated. A weighted sum of three values is used for any predicted luminance pixel value. This has the potential of generating a more accurate prediction and hence smaller prediction error.**

   This trend has continued — in H.264, the macroblock sizes can be even smaller, e.g., 8×4, 4×8, **or** 4×4.

10. In H.263 motion estimation, the *median* of the motion vectors from three preceding macroblocks (see Fig. 10.11(a)) is used as a prediction for the current macroblock. It can be argued that the median may not necessarily reflect the best prediction. Describe some possible improvements on the current method.

    Answer:
    Do more analysis: e.g., use the MV of the macroblock that has similar color and/or texture as the current macroblock; consistency with other MVs within the VOP, ...

11. H.263+ allows independent forward MVs for B-frames in a PB-frame. Compared to H.263 in PB-mode, what are the tradeoffs? What is the point in having PB joint coding if B-frames have independent motion vectors?

    Answer:
    As said in the text, H.263+ has independent forward motion vectors for B-Frames, just like MPEG, as opposed to H.263 which has P and B frames always together sharing the same motion vectors (joint coding). In PB joint coding you save bits on coding MVs assuming motion is consistent across 2 frames, sacrificing quality if not (thus needing higher bit rate for error coding); while in independent B-frame coding you lower prediction error if motion is jerkey yet you always have to code an additional set of MVs. The main point is to try to do better (MPEG-like) coding yet keeping compatibility with legacy concepts.

Chapter 11

MPEG Video Coding: MPEG-1, 2, 4 and 7

Exercises

1. As we know, MPEG video compression uses I-, P-, and B-frames. However, the earlier H.261 standard does not use B-frames. Describe a situation in which video compression would not be as effective without B-frames. (Your answer should be different from the one in Fig. 11.1.)

   Answer:

   Besides occlusion, the following could also call for bi-directional search: lighting (color and/or intensity) changes, changing views of 3D shape and/or texture, etc.

2. The MPEG-1 standard introduced B-frames, and the motion-vector search range has accordingly been increased from $[-15, 15]$ in H.261 to $[-512, 511.5]$. Why was this necessary? Calculate the number of B-frames between consecutive P-frames that would justify this increase.

   Answer:

   The range of $[-512, 511.5]$ is used for half-pixel precision. For full-pixel precision it is actually specified as $[-1, 024, 1, 023]$. Both are upper bounds, and may never be used. Note, the larger search window is not only due to the introduction of B-frames. In fact, it is partly made necessary because higher spatial resolution is now supported in MPEG-1 video frames.
   If we simply assume that B-frame is the only cause, then the calculation would suggest that up to $512/15 \approx 34$ B-frames could be in-between consecutive P-frames.

3. B-frames provide obvious coding advantages, such as increase in SNR at low bitrates and bandwidth savings. What are some of the disadvantages of B-frames?

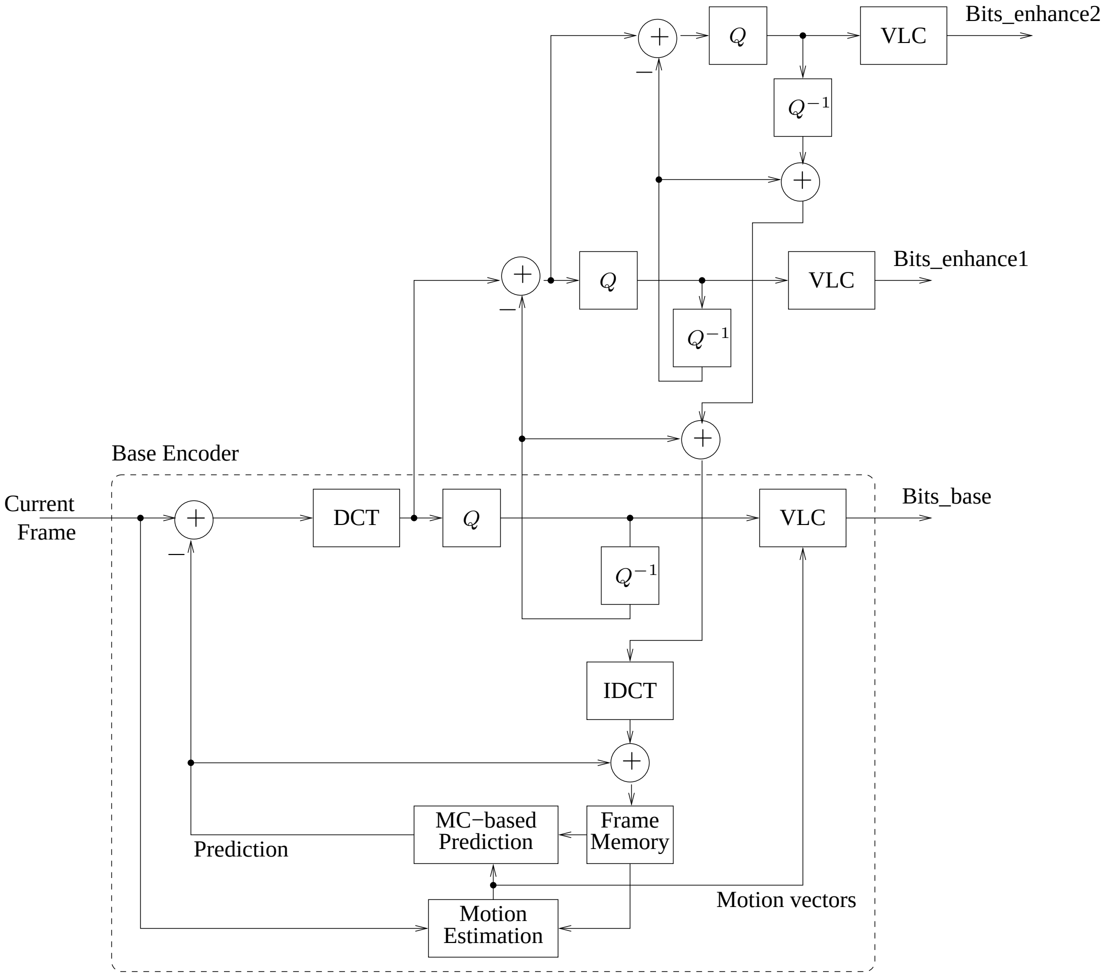
   Answer:

   The obvious one is the overhead — especially the time it needs to do bi-directional search at the encoder side.
   Also, it is hard to make a decision as how to use the results from both the forward and backward predictions: when there are fast motions/transitions, it can often be the case that only one of the motion vectors is accurate. Therefore, a simple average of the difference MBs may not yield good results.

4. Redraw Fig. 11.8 of the MPEG-2 two-layer SNR scalability encoder and decoder to include a second enhancement layer.

Answer:

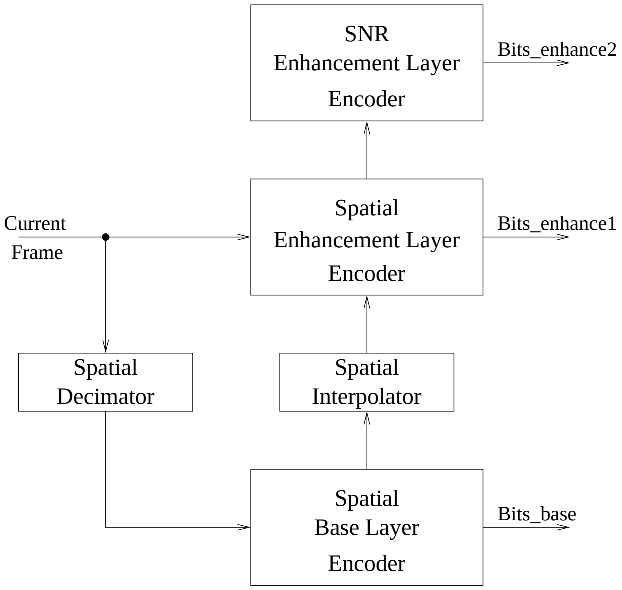
SNR scalability is based on different levels of quantization on the DCT coefficients of the difference blocks. The following is a three-level diagram.



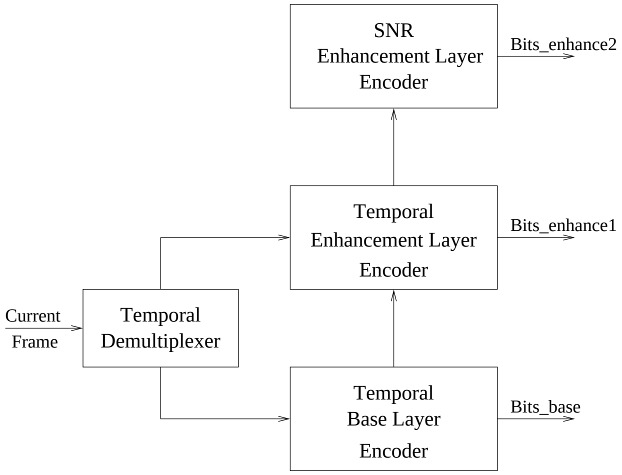
5. Draw block diagrams for an MPEG-2 encoder and decoder for (a) SNR and spatial hybrid scalability, (b) SNR and temporal hybrid scalability.

   Answer:

   (a) Block diagram for SNR and Spatial Hybrid Scalability:


SNR Enhancement Layer Encoder
Bits_enhance2
Current Frame
Spatial Enhancement Layer Encoder
Bits_enhance1
Spatial Decimator
Spatial Interpolator
Spatial Base Layer Encoder
Bits_base

   (b) Block diagram for SNR and Temporal Hybrid Scalability:


SNR Enhancement Layer Encoder
Bits_enhance2
Temporal Enhancement Layer Encoder
Bits_enhance1
Current Frame
Temporal Demultiplexer
Temporal Base Layer Encoder
Bits_base

6. Why aren't B-frames used as reference frames for motion compensation? Suppose there is a mode where any frame type can be specified as a reference frame. Discuss the tradeoffs of using reference B-frames instead of P-frames in a video sequence (i.e., eliminating P-frames completely).

   Answer:

   As is, the B-frame is at the lowest level of the I-P-B hierarchy. Any error made on P-frame motion compensation (MC) will be carried over to B-frame MC. Therefore, B-frames are not

used as reference frames.

In principle, this can be done (and in fact it is done in later standards.)

It depends on the quality of MC on B-frames. If, for example, B-frames are directly compared to preceding and succeeding I-frames, then the quality will be ensured. The only problem is that we need to introduce more I-frames in order to reduce the distance between B- and I-frames (hence to avoid too large a MV search range).

7. Write a program to implement the SNR scalability in MPEG-2. Your program should be able to work on any macroblock using any quantization *step_sizes* and should output both `Bits_base` and `Bits_enhance` bitstreams. The variable-length coding step can be omitted.

   Answer:
   This can be used as a Programming Assignment.

   Sample solution:

```
// MB is an object that can be accessed like array, stores 2D
// integers.
// RefImg is a reference frame for prediction (some image class
// structure, RefImg1 forward, RefImg2 backwards). void
SNRScalability(MB * pMB, int Q1step_size, int Q2step_size) {
  MB DCTMB, Q1DCTMB, Q2DCTMB, IQDCTMB, QMC;

  // Check for frame type, see if temporal prediction necessary.
  if((pMB->frameType == BFRAME) || (pMB->frameType == PFRAME)
  {
    // Calculate the prediction block (in QMB) from the reference
    //frames.
    MotionCompensate(QMB, pMB->MV, pMB->x, pMB->y,
        pMB->predictionType, RefImg1, RefImg2);
    *pMB -= QMB;
  }

  // DCT transform the Macroblock, put output in DCTMB.
  DCT(pMB, DCTMB);

  // Quantize the DCT base stream, and enhancement stream.
  for(int i=0; i<DCTMB.height; i++)
    for(int j=0; j<DCTMB.width; j++)
      Q1DCTMB[i][j] = DCTMB[i][j] * 8 / Q1step_size;
      Q2DCTMB[i][j] = (DCTMB[i][j] -
         Q1DCTMB[i][j] / 8 * Q1step_size) * 8 / Q2step_size;

  // output both bit streams at this point.
  Q1DCTMB.outputStream();
  Q2DCTMB.outputStream();

  // Calculate the prediction frame.
  IQDCTMB = Q1DCTMB/8*Q1step_size + Q2DCTMB/8*Q2step_size;
  IDCT(&IQDCTMB, QMB);

  // Check for image type to know how to update reference frame.
  if((pMB->frameType == IFRAME) || (pMB->frameType == PFRAME))
  {
    // Check for initial settings.
    if(RefImg1.NotInitialized())
    AddMBtoIMG(RefImg1, QMB);
    // We are assuming here that at when the frame changes
    // (to IFRAME or PFRAME)
    // the encoder copies RefImg2 to RefImg1.
    else
    AddMBtoIMG(RefImg2, QMB);
  }
}
```

8. MPEG-4 motion compensation is supposed to be VOP-based. At the end, the VOP is still divided into macroblocks (interior macroblock, boundary macroblock, etc.) for motion compensation.

   (a) What are the potential problems of the current implementation? How can they be improved?
   Answer:

   - motion vectors for various MBs with the same VOP may not be the same, sometimes they may not even be close.

   + easy, fast

   Possible improvement: Look around each MB in the VOP for continuity. Color, texture, shape (boundary) information can be used to assist in this process, i.e., using multiple cues.

   (b) Can there be true VOP-based motion compensation? How would it compare to the current implementation?
   Answer:

   This is the (3D) object-motion problem that computer vision researcher have been working on for more than three decades now.

   + motion vector closer to object motion, good for indexing, retrieval.

   - slower, could be wrong. The key is object segmentation which is error-prone.

9. MPEG-1 and 2, and 4 are all known as decoder standards. The compression algorithms, hence the details of the encoder, are left open for future improvement and development. For MPEG-4, the major issue of *video object segmentation*, i.e., how to obtain the VOPs, is left unspecified.

   (a) Propose some of your own approaches to video object segmentation.
   Answer:

   Traditional image segmentation methods rely on region growing (based on homogeneous color, texture, etc.), edge/contour detection, and the combination of both. They are known to be unreliable especially in presence of noise, shadow, lighting change.

Digital video (motion picture) has the added temporal (time) dimension. Temporal redundancy (similarity) can be explored to enhance the quality of object segmentation.

Since MPEG video comes with motion vectors for MBs, they can be used to aid object segmentation. Basically, MBs with similar motion vectors can be merged into larger segments. Color, texture, shape (boundary) information can be used to prevent any excessive merge. In other words, an iterative merge-and-split algorithm can be developed.

(b) What are the potential problems of your approach?

Answer:

Object is a difficult thing to recover. Certain features are inevitably lost or distorted in the process of video capturing. It is also not apparent how to incorporate high-level knowledge of objects, such as the presence and behavior of certain objects in certain scenes.

10. Motion vectors can have subpixel precision. In particular, MPEG-4 allows quarter-pixel precision in the luminance VOPs. Describe an algorithm that will realize this precision.

Answer:

Basically, the same bilinear interpolation method that generated the half-pixel image in Fig. 10.12 can be used one more time to generate quarter-pixel images. Namely, after obtaining half-pixels a, b, c, d as in Fig. 10.12, we can get the quarter-pixels a' = a, b' = (a+b+1)/2, ... This will be followed by the step of motion vector search, now at quarter-pixel precision.

As always, other interpolation methods and/or a larger neighborhood window can also be utilized.

11. As a programming project, compute the SA-DCT for the following 8×8 block:

$$\begin{array}{rrrrrrrr}
0 & 0 & 0 & 0 & 16 & 0 & 0 & 0 \\
4 & 0 & 8 & 16 & 32 & 16 & 8 & 0 \\
4 & 0 & 16 & 32 & 64 & 32 & 16 & 0 \\
0 & 0 & 32 & 64 & 128 & 64 & 32 & 0 \\
4 & 0 & 0 & 32 & 64 & 32 & 0 & 0 \\
0 & 16 & 0 & 0 & 32 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 16 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

12. What is the computational cost of SA-DCT, compared to ordinary DCT? Assume the video object is a 4×4 square in the middle of an 8×8 block.

Answer:

They are at the same order. Since SA-DCT operates on fewer input and output values, it is a bit more efficient.

Using the implementation of 2D separable DCT, the complexity of calculating each DCT coefficient is $O(N)$, where N is the block size. For all DCT coefficients in the block, the complexity is $O(N^3)$.

In the example above, ordinary DCT has a complexity of $O(8^3)$, whereas SA-DCT has $O(4^3)$.

13. Affine transforms can be combined to yield a composite affine transform. Prove that the composite transform will have exactly the same form of matrix (with $[0\ 0\ 1]^T$ as the last column) and at most 6 degrees of freedom, specified by the parameters a_{11}, a_{21}, a_{31}, a_{12}, a_{22}, a_{32}.

    Answer:

    This is guaranteed as long as the third column for each transform matrix is $[0, 0, 1]^T$. A general form is:

$$\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & 0 \\ b_{21} & b_{22} & 0 \\ b_{31} & b_{32} & 1 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & 0 \\ c_{21} & c_{22} & 0 \\ c_{31} & c_{32} & 1 \end{bmatrix},$$

    where $c_{11} = a_{11}b_{11} + a_{12}b_{21}$, and $c_{12} = a_{11}b_{12} + a_{12}b_{22}$, ...

14. Mesh-based motion coding works relatively well for 2D animation and face animation. What are the main problems when it is applied to body animation?

    Answer:

    Body is a 3D object and it is also non-rigid (deformable). Due to body motions, occlusion and deformation happen/change which can cause topology changes of the mesh.

15. What is the major motivation behind the development of MPEG-7? Give three examples of real-world applications that may benefit from MPEG-7.

    Answer:

    Indexing and retrieval of multimedia databases and digital libraries.

    Content-based image and video retrieval (broadcast programs and/or film archiving, museum and/or library paintings and books), e-commerce, tele-learning, tele-medicine, ...

16. Two of the main shape descriptors in MPEG-7 are "region-based" and "contour-based". There are, of course, numerous ways of describing the shape of regions and contours.

    (a) What would be your favorite shape descriptor?

    (b) How would it compare to ART and CSS in MPEG-7?

    Answer:

    (a) Our favorite is *locale* — the *feature localization* method that we described in Chapter 20.

    (b) CSS is invariant to translations and rotations, and it is pretty robust to scale changes. However, it relies on a good contour extraction which is (almost) non-attainable. ART is a set of descriptors at a pretty high level. It lacks the descriptive power (not quite usable).

    The locale-based descriptor is not based on "image segmentation". Instead, it attempts to localize features. Therefore it has a better chance to survive when a "segmentation" effort fails. When needed, locales can also have their own shape descriptors down to the pixel precision.

Chapter 12

New Video Coding Standards: H.264 and H.265

Exercises

1. Integer Transforms are used in H.264 and H.265.

   (a) What is the relationship between the DCT and Integer Transform?

   (b) What are the main advantages of using Integer Transform instead of DCT?

   Answer:

   (a) Each row of the Integer Transform matrix is an approximate and scaled down version from the DCT matrix.

   (b) No errors due to finite floating-point precision, so no drifting; also faster.

2. H.264 and H.265 use quarter-pixel precision in motion compensation.

   (a) What is the main reason that sub-pixel (in this case quarter-pixel) precision is advocated?

   (b) How do H.264 and H.265 differ in obtaining the values at quarter-pixel positions?

   Answer:

   (a) Higher precision in motion estimation, smaller residual errors.

   (b) After deriving half-pixel values, H.264 uses average from nearest pixels at half-pixel and/or integer positions; H.265 uses qfilters with 7 taps.

3. From Eq. 12.15, derive $\mathbf{H_{8\times8}}$ for the Integer Transform in H.265.

   Answer:

$$\mathbf{H_{8\times8}} = \begin{bmatrix} 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\ 89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 \\ 83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 \\ 75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 \\ 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 \\ 50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 \\ 36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 \\ 18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 \end{bmatrix} \tag{12.1}$$

4. H.264 and H.265 support *in-loop deblocking filtering*.

   (a) Why is deblocking a good idea? What are its disadvantages?

   (b) What are the main differences in its H.264 and H.265 implementations?

   (c) Beside the deblocking filtering, what does H.265 do to improve the visual quality?

   Answer:

   (a) Removing the blocky artifact. Over-smoothing, lost original sharp edges and color.

   (b) Block size: 8×8 **in H.265,** 4×4 **in H.264.**

   (c) SAO (Sample Adaptive Offset)

5. Name at least three features in H.265 that facilitate parallel processing.

   Answer:

   (a) Tiles, slices decoded independently

   (b) WPP (wavefront parallel processing)

   (c) Two-phase deblocking processing on 8×8 **edges only**

6. Give at least three reasons to argue that PSNR is not necessarily a good metric for video quality assessment.

   Answer:

   (a) Need for strict image alignment/scaling

   (b) Sensitive to color/intensity changes (e.g., histogram stretching)

   (c) Insensitive to structural loss

   (d) No support for any high-level/perceptual factors such as focus of attention, etc.

7. P-frame coding in H.264 uses *Integer Transform*. For this exercise, assume:

$$F(u,v) = H \cdot f(i,j) \cdot H^T, \text{ where } H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}.$$

   (a) What are the two advantages of using Integer Transform?

   (b) Assume the Target Frame below is a P-frame. For simplicity, assume the size of macroblock is 4×4. For the macroblock shown in the Target Frame:
   (i) What should be the Motion Vector?
   (ii) What are the values of $f(i,j)$ in this case? (iii) Show all values of $F(u,v)$.

20	40	60	80	100	120	140	155
30	50	70	90	110	130	150	165
40	60	80	100	120	140	160	175
50	70	90	110	130	150	170	185
60	80	100	120	140	160	180	195
70	90	110	130	150	170	190	205
80	100	120	140	160	180	200	215
85	105	125	145	165	185	205	220

Reference Frame

110	132	154	176	–	–	–	–
120	142	164	186	–	–	–	–
130	152	174	196	–	–	–	–
140	162	184	206	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–

Target Frame

Answer:

(a) No drifting; also faster.

(b) (i) MV = (3, 3).

(ii) $f(i,j)$ **are the differences.** $f(i,j) = \begin{bmatrix} 0 & 2 & 4 & 6 \\ 0 & 2 & 4 & 6 \\ 0 & 2 & 4 & 6 \\ 0 & 2 & 4 & 6 \end{bmatrix}$.

(iii) $F(u,v) = \begin{bmatrix} 48 & -56 & 0 & -8 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$.

8. Write a program for the kth Order Exp-Golomb encoder and decoder. (a) What is the EG_0 codeword for unsigned $N = 110$? (b) Given an EG_0 code 000000011010011, what is the decoded unsigned N? (c) What is the EG_3 codeword for unsigned $N = 110$?

   Answer:
   (a) 0000001101111. (b) 210. (c) 0001110110.

9. Write a program to implement video compression with motion compensation, transform coding, and quantization for a simplified H.26* encoder and decoder.

   - Use 4:2:0 for chroma subsampling.
   - Choose a video frame sequence (I-, P-, B-frames) similar to MPEG-1, 2. No interlacing.
   - For I-frames, implement the H.264 Intra_4×4 predictive coding.
   - For P- and B-frames, use only 8×8 for motion estimation. Use logarithmic search for motion vectors. Afterwards, use the 4×4 Integer Transform as in H.264.
   - Use the quantization and scaling matrices as specified in Eqs. 12.5 and 12.7. Control and show the effect of various levels of compression and quantization losses.
   - Do not implement the entropy coding part. Optionally, you may include any publicly available code for this.

10. Write a program to verify the results in Table 12.8. For example, to show that DST will produce shorter code than DCT for Category 2 directional predictions.

Chapter 13

Basic Audio Compression Techniques

Exercises

1. In Section 13.3.1 we discuss phase insensitivity. Explain the meaning of the term "phase" in regard to individual frequency components in a composite signal.

   Answer:
   The relative delay of individual frequency components.

2. Input a speech segment, using C or MATLAB, and verify that formants indeed exist — that any speech segment has only a few important frequencies. Also, verify that formants change as the interval of speech being examined changes.

   A simple approach to coding a frequency analyzer is to reuse the DCT coding ideas we have previously considered in Section 8.5. In one dimension, the DCT transform reads

   $$F(u) = \sqrt{\frac{2}{N}}\, C(u) \sum_{i=0}^{N-1} \cos \frac{(2i+1)u\pi}{2N}\; f(i), \tag{13.35}$$

   where $i, u = 0, 1, \ldots, N-1$, and the constants $C(u)$ are given by

   $$C(u) = \begin{cases} \frac{\sqrt{2}}{2} & if \quad u = 0, \\ 1 & otherwise. \end{cases} \tag{13.36}$$

   If we use the speech sample in Fig. 6.15, then taking the one-dimensional DCT of the first, or last, 40 msec (i.e., 32 samples), we arrive at the absolute frequency components as in Fig. 13.5.

   Answer:

```
% formants.m
% matlab script
load 'resources_exercises/chap13/noahwav.txt' -ascii; % 8000 samples.
% look for formants:
noahwav1 = noahwav(1:32); % beginning 40 msec
noahwav2 = noahwav(7969:8000);  % last 40 msec
formants1 = dct(noahwav1); % blocksize is same as noahwav1, i.e. 32.
formants2 = dct(noahwav2);
%
```

```
plot(abs(formants1));
hold on
plot(abs(formants2),’--’);
xlabel(’Frequency’);
ylabel(’abs( Coefficient)’);
hold off
```

3. Write code to read a WAV file. You will need the following set of definitions: a WAV file begins with a 44-byte header, in unsigned byte format. Some important parameter information is coded as follows:

Byte[22..23]	Number of channels
Byte[24..27]	Sampling Rate
Byte[34..35]	Sampling Bits
Byte[40..43]	Data Length

Answer:
A WAVE file is a RIFF file. The complete WAV format is at
http://www.wotsit.org/download.asp?f=wavecomp
The code for WavRead.cpp and WavRead.h can be downloaded from
http://www.microsoft.com/msdownload/platformsdk/Samples/Multimedia/DSound/Src
An older, 1999, version from Microsoft is at
http://140.131.13.205/LHU_EL_Teacher/el049/DirectX/DX7ASDK/DXF/samples/multimedia/dsound /src/voicemanagement/
and that version is copied into `resources_exercises/chap13/wavread.zip`**.**

4. Write a program to add fade in and fade out effects to sound clips (in WAV format). Specifications for the fades are as follows: The algorithm assumes a linear envelope; the fade-in duration is from 0% to 20% of the data samples; the fade-out duration is from 80% to 100% of the data samples.

   If you like, you can make your code able to handle both mono and stereo WAV files. If necessary, impose a limit on the size of the input file, say 16 megabytes.

   Answer:

```
% wavfade.m
% matlab script
wav = wavread(’resources_exercises/chap13/coffee.wav’);
wav = double(wav);
ss = size(wav,1); % 16000
fadesize = fix(ss/5);
fadefnc = ( (1:fadesize)/fadesize )’ ;
fadefnc2 = (fliplr(fadefnc’))’;
wavoutp = wav;
intro = wav(1:fadesize).*fadefnc;
outro = wav((ss-fadesize+1):ss).*fadefnc2;
wavoutp(1:fadesize) = intro;
wavoutp((ss-fadesize+1):ss) = outro;
plot(wav)
plot(wavoutp)
wavwrite(wavoutp, 'resources_exercises/chap13/coffeefade.wav');
```

5. In the text, we study an adaptive quantization scheme for ADPCM. We can also use an adaptive prediction scheme. We consider the case of one tap prediction, $\hat{s}(n) = a \cdot s(n-1)$. Show how to estimate the parameter a in an open-loop method. Estimate the SNR gain you can get, compared to the direct PCM method based on a uniform quantization scheme.

   Answer:
   To estimate a, we minimize the following function

   $$E[e^2] = E[(s(n) - as(n-1))^2]$$

   by $\partial E[e^2]/\partial a = E[(s(n) - as(n-1))(-s(n-1))] = 0$. The estimate of a is

   $$\begin{aligned} a &= E[s(n)s(n-1)]/E[s^2(n)] \\ &= R(1)/\sigma_s^2 \end{aligned}$$

   In ADPCM, $s(n) = e(n) + \tilde{s}(n) = e_q(n) + e_r(n) + \tilde{s}(n)$, where $\tilde{s}(n)$ is the prediction signal, $e_q(n)$ is the quantized difference signal of $s(n)$ and $\tilde{s}(n)$ and $e_r(n)$ is the quantization error of $e(n)$. Therefore, the quantization error in ADPCM is equal to the quantization error of $e(n)$. For uniform quantization, the variance of the quantization error is proportional to the variance of the input signal. Thus,

   $$\begin{aligned} E[e_r^2] \simeq E[e^2] &= E[(s(n) - as(n-1))^2] \\ &= \sigma_s^2(1 - a^2) \end{aligned}$$

   where $a = R(1)/\sigma_s^2$. Usually a is strictly less than 1. Thus the quantization SNR gain is $-10 \log 10(1 - a^2)$.

6. Linear prediction analysis can be used to estimate the shape of the envelope of the short-time spectrum. Given ten LP coefficients $a_1, \ldots, a_{10}$, how do we get the formant position and bandwidth?

   Answer:
   Solve the equation $1 + \sum_{i=1}^{10} a_i z^{-i} = 0$ and obtain the roots $\{z_i = r_i e^{j\theta_i}, i = 1, 2...10\}$. Then the normalized frequency of the ith formant is $F_i = \theta_i/(2\pi)$ and its bandwidth is $B_i = \log r_i/\pi$.

7. Download and implement a CELP coder (see the textbook web site). Try out this speech coder on your own recorded sounds.

8. In quantizing LSP vectors in G.723.1, splitting vector quantization is used: if the dimensionality of LSP is 10, we can split the vector into three subvectors of length 3, 3, and 4 each and use vector quantization for the subvectors separately. Compare the codebook space complexity with and without split vector quantization. Give the codebook searching time complexity improvement by using splitting vector quantization.

   Answer:
   If we use 8 bits for each subvector codebook, we need 24 bits for the LSP vector. The space used is pow(2, 8)* 10 * size(float). With general VQ, the space used is pow(2, 24) * 10 * size(float). The codebook searching time is proportional to the codebook size.

9. Discuss the advantage of using an algebraic codebook in CELP coding.

   Answer:
   Firstly, we do not need training. Second there, are many zeros and overlap between different codewords, so that fast codebook searching is possible.

10. The LPC-10 speech coder's quality deteriorates rapidly with strong background noise. Discuss why MELP works better in the same noisy conditions.

    Answer:
    LPC-10 uses a binary U/V decision, which can give a wrong decision and degrade the synthesized speech. MELP uses a multi-model U/V decision. In fact, it uses a voice degree to describe the ratio of noise and periodic components in each band. This can give a much better fit to real, noisy, input speech.

11. Give a simple time-domain method for pitch estimation based on the autocorrelation function. What problem will this simple scheme have when based on one speech frame? If we have three speech frames, including a previous frame and a future frame, how can we improve the estimation result?

    Answer:
    Find the peaks of the autocorrelation function. If we have multi-frames we can use dynamic programming and delay the decision.

    That is, in the multi-frame situation we can choose a pitch based not only on the autocorrelation function from a single frame but instead based on several frames by adding a smoothness constraint. For example, we can choose 10 local peaks of the autocorrelation function as the pitch candidates in the current frame and in the future frame. Since the pitch in the previous frame is already determined, our task now is to find a solution (three pitches) in the three successive frames which optimizes some cost function. One naive solution is by exhaustively searching 10×10=100 possibilities. A better method involves dynamic programming.

    A student solution includes defining some energy function, and there are many possible candidates.

12. On the receiver side, speech is usually generated based on two frames' parameters instead of one, to avoid abrupt transitions. Give two possible methods to obtain smooth transitions. Use the LPC codec to illustrate your idea.

    Answer:
    One method is interpolating the parameters for consecutive frames and forming a time-varying synthesis filter to process the excitation. The second method synthesizes the speech in each frame, separately and overlapping, with addition by some appropriate windowing.

Chapter 14

MPEG Audio Compression

Exercises

1. (a) What is the threshold in quiet, according to Eq. (14.1), at 1,000 Hz? (Recall that this equation uses 2 kHz as the reference for the 0 dB level.)

   Answer:
   The threshold in quiet is:

   $$3.64 * (f/1000)^{-0.8} - 6.5 * exp(-0.6 * (f/1000 - 3.3)^2) + 10^{-3} * (f/1000)^4$$

   At $f = 1000$, this evaluates to 3.369067.

   (b) Take the derivative of Eq. (14.1) and set it equal to zero, to determine the frequency at which the curve is minimum. What frequency are we most sensitive to? Hint: One has to solve this numerically.

   Answer:

```
# Maple script:
tiq := 3.64*f^(-0.8) - 6.5*exp( (-0.6)*(f-3.3)^2 ) + 10^(-3)*f^4;
# f in kHz
df:=diff(tiq,f);
fsolve(df,f);
# 3.324133041
== 3324.133041 Hz
```

2. Loudness versus Amplitude. Which is louder: a 1,000 Hz sound at 60 dB or a 100 Hz sound at 60 dB?

   Answer:
   The 1000 Hz sound at 60 dB will have loudness at a phon level of 60dB.
   The 100 Hz sound will be perceived as a phon level of approximately 50 dB so the 1000 Hz sound will be much louder.

3. For the (newer versions of the) Fletcher-Munson curves, in Fig. 14.1, the way this data is actually observed is by setting the y-axis value, the sound pressure level, and measuring a human's estimation of the effective perceived loudness. Given the set of observations, what must we do to turn these into the set of perceived loudness curves shown in the figure?

Answer:
Invert the functions — just use linear interpolation. In detail, according to the Robinson-Dadson paper (and others), we can capture the received loudness (phons, P) as a function of the stimulus D (also in dB) via a function $P = a + bD + cD^2$, where a,b,c are functions of frequency f. Thus for each f we have a set of values P(D), with D=0:10:120, say. Now we'd like to develop a second set of values, D=D(P), for equally-spaced Phon values P. So we interpolate. The interpolation is unworkable on the very low and very high D ends, unless we use better than linear interpolation.

4. Two tones are played together. Suppose tone 1 is fixed, but tone 2 has a frequency that can vary. The *critical bandwidth* for tone 1 is the frequency range for tone 2 over which we hear *beats*, and a roughness in the sound. Beats are overtones at a lower frequency than the two close tones; they arise from the difference in frequencies of the two tones. The critical bandwidth is bounded by frequencies beyond which the two tones sound with two distinct pitches.

   (a) What would be a rough estimate of the critical bandwidth at 220 Hz?

   Answer:
   According to eq. (14.5), the critical bandwidth (df) for a given center frequency f can also be approximated by

   $$df \;=\; 25 \;+\; 75 \times [1 + 1.4(f^2)]^{0.69} \quad ,$$

   where f is in kHz and df is in Hz.

   According to this formula, at 0.22 kHz, the bandwidth is roughly 103.5 Hz.

   (b) Explain in words how you would set up an experiment to measure the critical bandwidth.

   Answer:
   Dr. David Brainard (in `SoundThreshWriteup.pdf`) writes: "The idea behind critical bands is that sounds at different frequencies are processed by different auditory channels or mechanisms. If this is so, then a masker at one frequency will not mask a test at another frequency that is sufficiently different, since if the two differ in frequency they will be processed by different channels. To test this, you would measure the threshold to detect a tone in the presence of a masker at different frequencies. For this experiment, it is a good idea to set the test bandwidth to zero and the masker bandwidth to about 200 or so. You then measure the threshold T as a function of the masker frequency F. There should be a lot of masking (a high threshold) when masker has the same frequency as the test. But as the frequency of the masker increases or decreases from the test, we expect the threshold for the test to get smaller, producing an inverted U-shaped curve when T is plotted against F. The width of the U-shape curve tells us the range of frequencies that the mechanism detecting the test is sensitive to. This critical bandwidth is thought to increase with test frequency, and a really nice experiment would be to try to measure this increase in bandwidth."

5. Search the web to discover what is meant by the following psychoacoustic phenomena:

   (a) Virtual Pitch

   Answer:
   A pure sine wave has so-called "spectral pitch" that is easy to comprehend. In contrast, "virtual pitch" refers to the basic pitch a himan senses for a complex harmonic tone. In this case, there is not any "actual" specific pitch, but just a general one that is perceived—it is not really there, so is "virtual". An example is male speech, which is perceived as a

bass tone rising and falling, notwithstanding the fact that there is a complex wave form actually present.

A more complex version of this is that if several harmonics are present, but the fundamental is filtered out, one can still perceive the sound as belonging to the fundamental, rather than the harmonics that are in fact present! The sound perceived in this case depends on how the harmonics are spaced out: the wider the spacing between harmonics that are allowed to remain in the signal, the higher is the virtual pitch perceived.

(b) Auditory scene analysis

Answer:

"a process in which the auditory system takes the mixture of sound that it derives from a complex natural environment and sorts it into packages of acoustic evidence in which each package probably has arisen from a single source of sound."

In particular, auditory scene analysis is concerned with computer modeling of the process by which humans convert complex sound into distinct, interpreted abstractions such as the words of a particular speaker. The computational problem often comes down to separating speech from interfering noise.

(c) Octave related complex tones

Answer:

Ascending or descending "staircases" of tones make us perceive an ambiguity in the pitch: these "octave-related complex tones" are in fact perceived by us via by following the pitch that is moving. The ambiguity is to ascertain just what octave to place a sound into, if several harmonics are present at the same time. Our way out of the problem is to simply choose the moving tone, if there is one. Shepard Scales consist of such harmonics, that include a "staircase" – and that is the tone that one automatically hears best.

(d) Tri-tone paradox

Answer:

This phenomenon again has to do with octave related sinewave components. Whereas Shepard tones consist of 10 components (octaves) with magnitudes weighted by a Gaussian over log-frequency, the tones used in the tritone paradox have 6 components (octaves) with weights given by a filter that cuts off more sharply, again over log-frequency.

Here, pairs of tones are played together. When the second tone is less than one half an octave (an interval called the "tritone" — e.g., C to F#) above the first, we hear the second tone as higher in pitch compared to the first (e.g., C to E is less than a tritone). But if the second tone is more than a tritone above the first, (e.g., C to G is more than a tritone) we perceive the second as lower in pitch compared to the first.

(e) Inharmonic complex tones

Answer:

Another paradox. Here we listen to a set of non-harmonics: tones that are separated by some particualr, but not special, frequency difference. First we hear a set of six tones each separated by the same difference, and then the whole set is moved up in frequency, by a small step. At some point,at the top of the frequency range represented, the whole set is re-initialized back where it started and the march upwards recommences. The result: we insist on hearing the higher tones, i.e., we hear the staricase of sound go on ascending even though in fact it has been re-initialized.

6. If the sampling rate f_s is 32 ksps then, in MPEG Audio Layer 1, what is the width in frequency of each of the 32 subbands?

Answer:
500 Hz.

7. Given that the level of a *masking tone* at the 8th band is 60 dB, and 10 msec after it stops, the masking effect to the 9th band is 25 dB.

   (a) What would MP3 do if the original signal at the 9th band is at 40 dB?

   (b) What if the original signal is at 20 dB?

   (c) How many bits should be allocated to the 9th band in (a) and (b) above?

   Answer:
   Only send 40-25 = 15 dB.
   Send 3 bits instead of 7 bits → saving of 4 bits.
   Send no bit.

8. What does MPEG Layer 3 (MP3) audio do differently from Layer 1 to incorporate temporal masking?

   Answer:
   More (1,152) samples per frame. Includes psychoacoustic model with temporal masking effects.

9. Explain MP3 in a few paragraphs, for an audience of consumer-audio-equipment salespeople.

   Answer:
   MPEG audio layer 3 is a type of audio codec for achieving significant compression from the original audio source with very little loss in sound quality. A compression ratio of up to 12:1 produces very little degradation. The standard bit rate ("near-CD" quality) is 128 or 112 kbit/s. An advantage of MP3 is that files can be broken up into pieces, with each piece is still playable. The feature that makes this possible (headerless file format) also means that MP3 files can be streamed in real-time. A disadvantage of MP3 compression is that high processor power is required to encode and play files in software. Hardware player/encoder/decoders are still quite expensive.

10. Implement MDCT, just for a single 36-sample signal, and compare the frequency results to those from DCT. For low-frequency sound, which does better at concentrating the energy in the first few coefficients?

    Answer:
    Solutions for forward and inverse MDCT, using straightforward loops and also using a more vector-based matlab approach, and in `resources_exercises/chap14/` **as** `mdctl.m`, `imdctl.m`**,** `mdctv.m`**, and** `imdctv.m`**.**

    Suppose we take a small segment of sound, as follows: Then we find, below, that the DCT does a considerably better job at concentrating the energy for a general signal. However, while not placing energy in a single first coefficient, the MDCT does much better overall in concentrating the energy in the first few coefficients for a *smooth* signal.

```
% matlab script
sig = rand(8); % uniform random, 8x8 matrix
sig = sig(:,1); % 8x1
% 0.5869 0.0576 0.3676 0.6315 0.7176 0.6927 0.0841 0.4544
% sig = [0.5869 0.0576 0.3676 0.6315 0.7176 0.6927 0.0841
%        0.4544]';
```

```
sigd = dct(sig);
% 1.2701 -0.0447 -0.3180 0.2411 0.4202 -0.0177 0.3654 -0.0726
% 1st component large compared to others.
abs(sigd(1))/max(sigd(2:end)) % 3.0229
sigm = mdctv(sig);
% -1.8215 -0.2013 0.4515 0.7666
abs(sigm(1))/max(sigm(2:end)) % 2.3764
%
% and on a bigger set:
%
sig = rand(100); % uniform random, 100x100 matrix
sig = sig(:); % 10,000x1
sigd = dct(sig); % length==10,000
abs(sigd(1))/max(sigd(2:end)) % 46.6140
sigm = mdctv(sig); % length==5,000
abs(sigm(1))/max(sigm(2:end)) % 5.2018
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Low-freq:
% on a small low-freq signal:
%
sig = randn(36); % gaussian random, 36x36 matrix
sig = sig(:,1); % 36x1
sig = sort(sig);
sigd = dct(sig); % length==36
abs(sigd(1))/max(sigd(2:end)) % 3.8880
sigm = mdctv(sig); % length==18
abs(sigm(1))/max(sigm(2:end)) % 4.7657
%
sig = randn(100); % gaussian random, 100x100 matrix
sig = sig(:); % 10,000x1
sig = sort(sig);
plot(sig); % smooth....
sigd = dct(sig); % length==10,000
abs(sigd(1))/max(sigd(2:end)) % 4.9593
abs(sigd(1))/max(sigd(20:end)) % 16.7562
sigm = mdctv(sig); % length==5,000
abs(sigm(1))/max(sigm(2:end)) % 3.9110
abs(sigm(1))/max(sigm(20:end)) % 87.1210
```

11. Convert a CD-audio cut to MP3. Compare the audio quality of the original and the compressed version — can you hear the difference? (Many people cannot.)

    Answer:
    The answer depends on the compression level set. According to
    http://www.audioboxinc.com/quality.html,
    “A majority of listeners can distinguish the difference between an MP3 audio file, at any compression level, and an uncompressed audio file. A majority of listeners can hear quality loss

from a TAPE RECORDING of an MP3 file at any compression level higher than 10:1.”

12. For two stereo channels, we would like to be able to use the fact that the second channel behaves, usually, in a parallel fashion to the first, and apply information gleaned from the first channel to compression of the second. Discuss how you think this might proceed.

    Answer:
    One simple approach is to encode one channel, plus the sum L+R. In terms of MPEG Audio, we would then assign independent left- and right-channel scalefactors.

Chapter 15

Network Services and Protocols for Multimedia Communications

Exercises

1. What is the main difference between the OSI and TCP/IP reference models? Describe the functionalities of each layer in the OSI model and their relations to multimedia communications.

   Answer:

   The ISO OSI reference model lists 7 logically separate layers to be implemented for a complete network connectivity. Each layer can be independently implemented by a different source in accordance with any technology, and they must all inter-operate according to the OSI specifications. In contrast, the TCP/IP protocol stack was developed as a practical solution to interfacing separate networks. The TCP/IP stack has fewer layers, thus easier to implement, and the application layer was developed separately on top of the protocol suite itself.

   (a) Physical Layer. Defines the electrical and mechanical properties of the physical interface (e.g., signal level, specifications of the connectors, etc.); also specifies the functions and procedural sequences performed by circuits of the physical interface.

   (b) Data Link Layer. Specifies the ways to establish, maintain, and terminate a link, such as the transmission and synchronization of data frames, error detection and correction, and access protocol to the Physical layer.

   (c) Network layer. Defines the routing of data from one end to the other across the network, using circuit switching or packet switching. Provides such services as addressing, internetworking, error handling, congestion control, and sequencing of packets. Quality-of-service routing or resource reservation in this layer can facilitate multimedia communications.

   (d) Transport layer. Provides end-to-end communication between *end systems* that support end-user applications or services. Supports either *connection-oriented* or *connectionless* protocols. Provides error recovery and flow control. Real-Time Protocol (RTP) for multimedia communications resides in this layer.

   (e) Session layer. Coordinates the interaction between user applications on different hosts, manages sessions (connections), such as completion of long file transfers or a multimedia conversation.

   (f) **Presentation layer. Deals with the syntax of transmitted data, such as conversion of different data formats and codes due to different conventions, compression, or encryption, which are common in multimedia communications.**
   (g) **Application layer. Supports various application programs and protocols, including multimedia applications.**

2. True or False.

   (a) ADSL uses cable modem for data transmission.
   (b) To avoid overwhelming the network, TCP adopts a flow control mechanism.
   (c) TCP flow control and congestion control are both window based.
   (d) Out-of-order delivery wont happen with Virtual Circuit.
   (e) UDP has lower header overhead than TCP.
   (f) Datagram network needs call setup before transmission.
   (g) The current Internet does not provide guaranteed services.
   (h) CBR video is easier for network traffic engineering than VBR video.

   Answer:

   F F T T T F T T

3. Consider multiplexing/demultiplexing, which is one of the basic functionalities of the transport layer.

   (a) List the 4-tuple that is used by TCP for demultiplexing. For each parameter in the 4-tuple, show a scenario that the parameter is necessary.
   (b) Note that UDP only uses the destination port number for demultiplexing. Describe a scenario where UDP's scheme fits better. *Hint: The scenario is very common in multimedia applications.*

   Answer:

   (a) Source port, source IP address, destination port, destination IP address. The source port and IP address are necessary for the receiver to reply to the sender, e.g., in a client/server transaction, when the server replies a request initiated by a client (the sender). The destination port is necessary when the receiver runs different applications (e.g., port 80 for Web and port 25 for email (SMTP)). The destination IP address is necessary if the receiver has multiple IP addresses.
   (b) UDP's scheme is connection-less, which is of lower overhead and works better for a multicast scenario in which each participant expects to hear others, e.g., in a multi-party video conference, because there is no need to maintain separate connections for each pair of sender/receiver.

4. Find out the IP address of your computer or smartphone/tablet. Is it a real physical IP address or an internal address behind a NAT?

   Answer:

   For windows system, the IP address can be found using command line `ipconfig /all`**.**

5. Consider a NAT-enabled home network. (a) Can two different local clients access an external web server simultaneously? (b) Can we establish two web servers (both of port 80) in this network, which are to be accessed by external computers with the basic NAT setting? (c) If we want to establish only one web server (with port 80) in this network, propose a solution and discuss its potential problems.

   Answer:

   (a) Yes
   (b) No
   (c) One possible solution is to use *port forwarding*, which configures the NAT device to send all packets received on a particular port to a specific host in the internal network.

6. What is the key difference between IPv6 and IPv4, and why are the changes in the IPv6 necessary ? Note that the deployment of IPv6 remains limited now. Explain the challenges in the deployment and list two interim solutions that extend the lifetime of IPv4 before IPv6 is fully deployed.

   Answer:
   The key difference is that IPv6 supports 128bit IP addresses, as opposed to the 32bit address field length in IPv4. Other differences include extended supports to multicasting, security and header usage. Additional flow control and QoS support is achieved using a labelling of packets that have other priorities than default. Multicasting is supported by adding a "scope" field to the IPv6 header, and a new address called "anycast address" is defined to indicate sending a packet to all group nodes.

   The challenge in deployment is mainly because the Internet is of a ultra-large scale and is an open interconnected network with no central controller. As such, the transition from IPv4 to v6 cannot be done instantly, and the transition is known to be very difficult given the co-existence of the two IP versions.

   Some interim solutions that make more effective use of the IPv4 addresses (and hence extend its lifetime): (1) Network Address Translation (NAT); (2) Classless Inter-Domain Routing (CIDR); (3) Dynamic Host Configuration Protocol (DHCP).

7. Discuss the pros and cons of implementing multicast in the network layer or in the application layer. Can we implement multicast in any other layer, and how?

   Answer:

   Multicast in the network layer is efficient in routing and allows open and anonymous membership; it however needs change to the router implementation, which is known to be very difficult in the global Internet; application-specific operations can be difficult to implement as well.

   Multicast in the application layer is easy to implement and deploy, and also easy to accommodate application-specific operations; it however is sub-optimal in routing (can introduce redundant traffic).

   Since the data link layer or physical layer deals with local transmission only, they cannot be used for global-network multicast. The transport layer can be used for multicast though, e.g.,

the source sends a data stream to each individual receiver (that is, using multiple unicast to implement multicast).

8. What is the relation between delay and jitter? Describe a mechanism to mitigate the impact of jitter.

   Answer:

   Delay is measured as the time needed from transmission to reception, whereas jitter is related to the variance of delays.

   A buffer is often used to hold a batch of incoming frames and output them with regular intervals, so as to reduce playback jitter.

9. Discuss at least two alternative methods for enabling QoS routing on packet-switched networks based on a QoS class specified for any multimedia packet.

   Answer:
   Assume the packets in the network all have a class specifier to indicate the level of QoS required by the users for the specific payload.

   The first possible routing methodology is to have routers sort the packets as they arrive according to the priority determined jointly by QoS required and the amount of time in the queue. The packets will be routed to their destination in the sorted order with the first packet in the queue dispatched first.

   Another possible routing method though less likely is to classify network trunks according to low/high capacity/delay/channel error and forward packets as they come but through routes that are appropriate to the QoS requested so that high bandwidth/low delay channels won't get bogged down with low priority traffic. Such routing information can be maintained dynamically and at every router in the network.

10. Consider the ATM network and today's Internet.

    (a) What are the key differences between the two types of networks? Why does the Internet become the dominating network now?

    (b) What are the key challenges for multimedia over the Internet?

    Answer:

    (a) Internet: datagram, variable packet length, best-effort service;
    ATM: virtual circuit, fixed cell length, guaranteed or semi-guaranteed services.

    (b) No service guarantee (in terms of bandwidth, delay, jitter, and etc.).

11. Consider the *additive increase and multiplicative decrease* (AIMD) congestion control mechanism in TCP.

    (a) Justify that AIMD ensures fair and efficient sharing for TCP flows competing for bottleneck bandwidth.

    To facilitate your discussion, you may consider the the simplest case with two TCP users competing for a single bottleneck. In Fig. 15.1, the throughputs of the two users are represented by

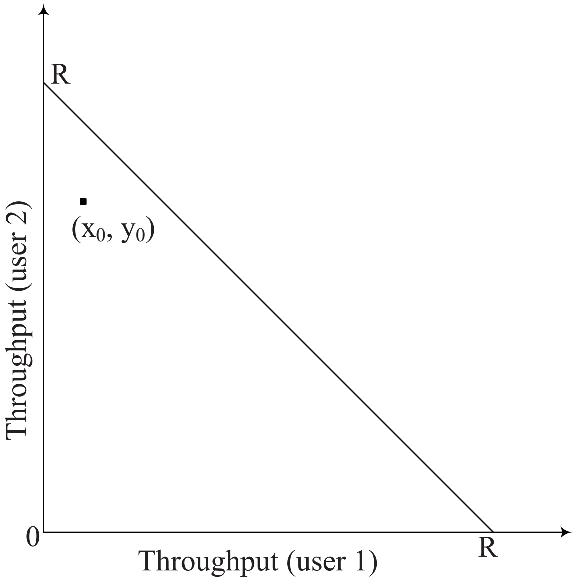


Fig. 15.1: Throughput of two TCP users sharing a bottleneck.

the X-axis and the Y-axis, respectively. When the aggregated throughput exceeds the bottleneck bandwidth R, congestion will happen (in the upper right side of the figure), though it will be detected after a short delay given that TCP uses packet loss as the congestion indicator.

For an initial throughput of the two users, say, x_0 and y_0, where $x_0 < y_0$, you can trace the their throughput change with AIMD, and show that they will eventually converge to a fair and efficient share of the bottleneck bandwidth. *Hint: There is only one such point.*

(b) Explain whether AIMD is suitable for multimedia streaming applications or not.

(c) Explain the relation between AIMD and TCP Friendly Rate Control (TFRC).

Answer:

(a) See Figure 15.2. The star is the optimal share at the center of the figure.

(b) The bitrate of a multimedia stream (e.g., video and audio) generally depends on its own content and often has a lower threshold, and therefore AIMD does not work well.

(c) AIMD in TCP is window-based and the transmission rate fluctuates greatly (saw-tooth). TFRC tries to achieve a similar average bandwidth as if a AIMD-based TCP connection is running over the same path; its transmission rate control however is equation-based and targets long-term equivalence with less short-term fluctuations.

12. TCP achieves reliable data transfer through re-transmission.

(a) Discuss the possible overheads of re-transmission.

(b) List two applications that re-transmissions are necessary.

(c) List two applications that re-transmissions are not necessary or not possible. Explain your answer.

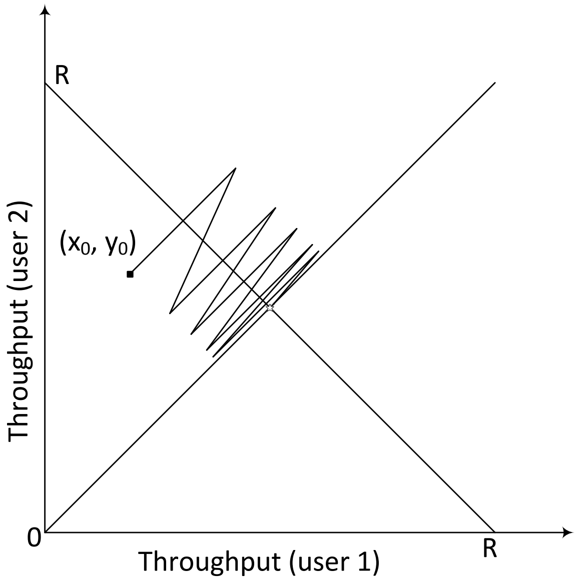


Fig. 15.2: Throughput of two TCP users sharing a bottleneck with throughput evolution.

Answer:

(a) More traffic. Longer delay. Blocked connection.

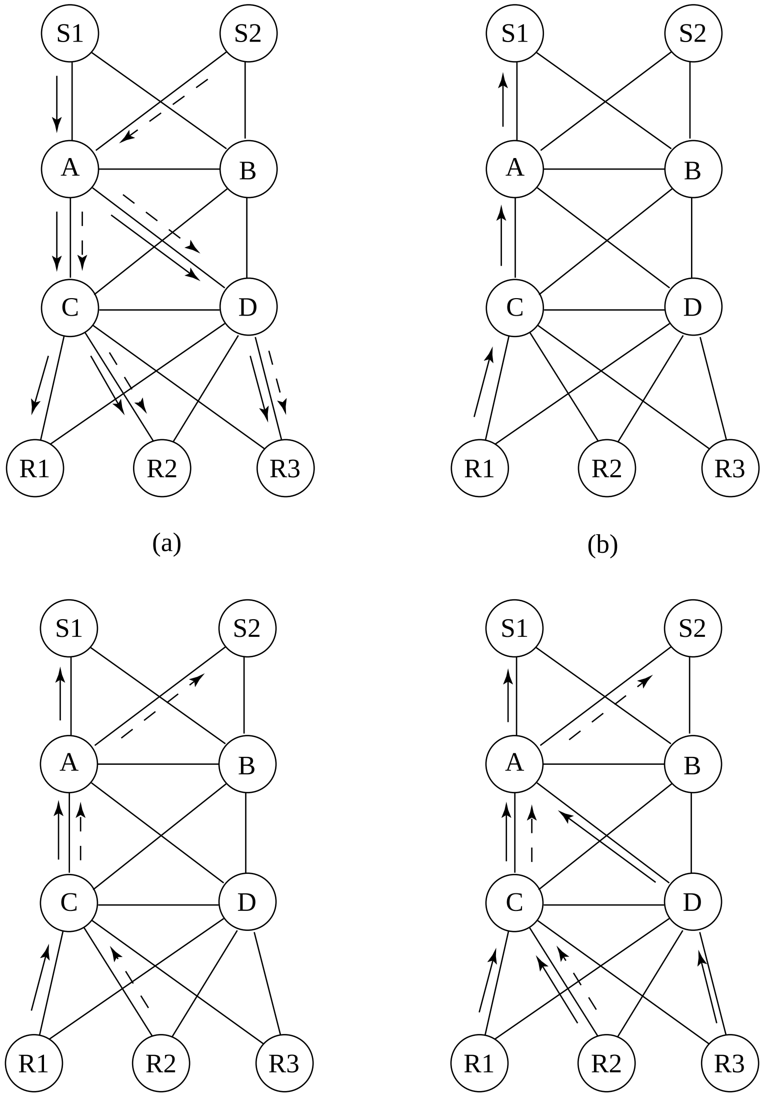
(b) E.g., Email and HTTP (Web).

(c) E.g., online gaming and two-way voice conversation (very time sensitive; can tolerant certain loss); or if the underlying network is highly reliable (e.g., a fiber network with almost no error), then retransmission is not necessary (to combat the rare errors, forward error correction (FEC) could be used; see Chapter 17).

13. Explain why RTP does not have a built-in congestion control mechanism, while TCP does. Also note that RTSP is independent of RTP for streaming control, i.e., using a separate channel. This is known as *out-of-band*, because the data channel and control channel are separated. Are there any advantage or disadvantage in combining both of them into a single channel?

Answer:
RTP targets diverse multimedia applications, which can have quite different demands. For example, a voice conversation may want to have a constant bitrate for transmission, whereas a live video streaming service may adjust the encoding rate in realtime to adapt to changing network conditions. A single built-in congestion control mechanism in RTP cannot work for all. On the other hand, TCP targets a set of applications of common demands: reliable, delay-insensitive, and bandwidth-insensitive. A single congestion control mechanism based on AIMD therefore works.

Combing the data and control channels eliminates a dedicated channel (a separate connection) for control, thus incurring lower overhead. However, this can make the control less responsive if they are mixed with bulk data transfers.



(a)

(b)

(c)

(d)

Fig. 15.3: A scenario of network resource reservation with RSVP: (a) senders S1 and S2 send out their `PATH` messages to receivers R1, R2, and R3; (b) receiver R1 sends out `RESV` message to S1; (c) receiver R2 sends out `RESV` message to S2; (d) receivers R2 and R3 send out their `RESV` messages to S1.

14. Consider Figure 15.3 that illustrates RSVP. In (d), receiver R3 decides to send an RSVP `RESV` message to S1. Assuming the figure specifies the complete state of the network, is the path reserved optimal for maximizing future network throughput? If not, what is the optimal path? Without modifying the RSVP protocol, suggest a scheme in which such a path will be discovered and chosen by the network nodes.

    Answer:
    Although the path that R3 reserves is the one sent by the PATH message from S1, it reserves 2 unused links before the stream merges at node A. A more optimal path (assuming equal link capacity naturally) would be R3 to C and merge there, thus reserving only one additional link and saving bandwidth for future streams.

A more optimal path could be established by the RSVP servers when they send the PATH message. A possible scheme can be to choose a link with the highest remaining capacity as the main trunk for transmission and branch off from it to clients at the points that minimize a measure combining the number of hops to the client and the capacity of such links. The idea would be to allow as many as possible future reservations.

15. Consider a typical Internet telephony system of 64 kbps data rate with a sampling frequency of 8 KHz.

    (a) If the data chunks are generated every 20 ms, how many data samples are there in each data chunk, and what is the size of each chunk?

    (b) What is the header overhead when a data chunk is encapsulated into the RTP/UDP/IP protocol stack.

    (c) Assume there is only one caller and one callee, what is the bandwidth allocated to RTCP?

    Answer:

    (a) 160 samples/chunk; 1280 bits/chunk (these are calculated without considering the Ethernet/IP/UDP/RTP header overhead; if the overhead is considered, the value will be smaller).

    (b) 3.37Kbps (RTCP bandwidth usage should generally not exceed 5% of total session bandwidth).

16. **Specify on Figure 15.13 the characteristics of *feasible* video transmission schedules. What is the *optimal transmission schedule*?**

Answer:
See Figure 15.4 (Exercise-answer). The grayed out area is the space where all feasible transmissions can exist, while the solid line is the optimal transmission schedule to minimize rate variability without prefetching any data first.

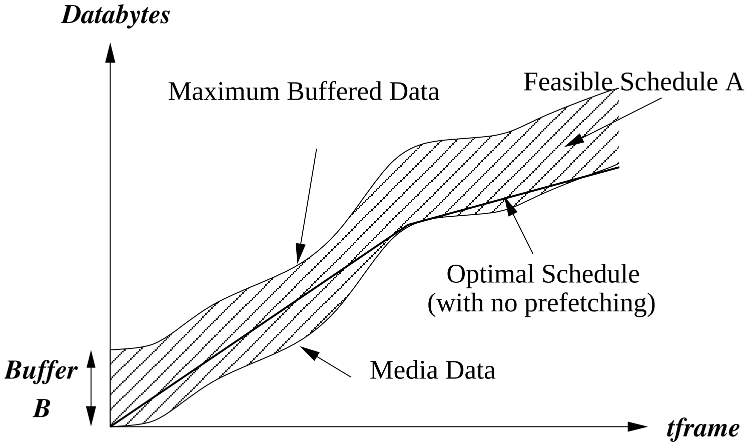


Fig. 15.4: Exercise-answer.

Chapter 16

Internet Multimedia Content Distribution

Exercises

1. Consider prefix caching with a total proxy cache size of S and N videos of user access probabilities of $r_1, r_2, ..., r_N$, respectively. Assume that the *utility* of caching for each video is given by function $U(l_i)$ where l_i is the length of the cached prefix for video i. Develop an algorithm to optimize the total utility of the proxy. You may start from the simple case where $U(l_i) = l_i \cdot r_i$.

   Answer:

   In the scheduling problem we consider here, we have proxy servers that can cache videos, and we have a set of videos $\{1, 2, ..., n\}$, and each has a given user access probability $r_1, r_2, ..., r_N$, respectively. Each video i can be prefix caching with the length l_i. We are also given a bound S and s denotes available caching size the in proxy server. We also would like to select a subset $M \subset N$ of the videos so that $\sum_{i \in M} l_i \leq S$ and, subject to this restriction, $\sum_{i \in M} l_i$ is as large as possible.

   Then we can show how to use dynamic programming to solve this problem. Let OPT(i) denotes the maximum total utility of proxy. We can denote $OPT(0) = 0$, then compute $OPT(i)$ in the order of $i = 1, ..., n$. In each step, the algorithm finds
   If $s < l_i$ then $OPT(i, s) = OPT(i-1, s)$. Otherwise

   $$OPT(i, s) = \underset{1 \leq i < N}{max} \{OPT(i-1, s), U(l_i) + OPT(i-1, s-l_i)\}\} \qquad (16.1)$$

   Finally, it returns $OPT(n)$.

2. For the optimal work-ahead smoothing technique, how would you algorithmically determine at which point to change the planned transmission rate? What is the transmission rate?

   Answer:
   Suppose we are looking at frame q of the video. We know the segment start point is frame $p+1$. We can detect overflow when the minimum possible rate R_{min} over the interval $p+1-q$ is higher than the rate to fully fill the buffer at time q, that is when:

   $$R_{min} > \frac{W(q) - (D(p) + B(p))}{q - p}$$

Note: **At the last frame we have nothing more to buffer, hence is $q = N$. We also consider it the end of the segment and use R_{min} over it. Since an overflow occurred at the minimum rate, we must reduce the rate at the last point q' where R_{min} was achieved (was exactly equal to it). The new rate segment is then from frames $p+1-q'$ at rate R_{min}. We set $p = q'$ for the next segment.**

Similarly, to know where an underflow occured we test for

$$R_{max} < \frac{D(q) - (D(p) + B(p))}{q - p}$$

If indeed the maximum possible rate over the interval is less than the minimum necessary to play the stream then we must increase the rate at last frame q' where R_{max} was achieved. The transmission rate for the segment is naturally R_{max}.

3. Considering again the optimal work-ahead smoothing technique, it was suggested that instead of using every video frame, only frames at the beginning of statistically different compression video segments can be considered. How would you modify the algorithm (or video information) to support that?

   Answer:
   Here, we must assume the video is coded in a way that finds significant points of compression change, presumably scene changes or larger motion, and so on. Let's assume the compressed video has a way of indicating this for example by assigning new I-frames to only such frames or starting a new GOP every time a the coder adapts to a new sequence. We can now create a new virtual frame sequence S' where the frames are only the ones indicated by the compressed video as a point of change. However, instead of specifying the frame size as in the source video, the frame size is the total amount of data in the video up to the frame (including the frame) minus the previous frame size in the virtual sequence. This gives only the amount of data needed to transmit up to the compression change frame, and we can safely assume that this data can be transmitted at constant bit-rate, whatever rate it is as calculated by the work-ahead smoothing algorithm.

   The technique is applied to sequence S' instead of the original video and so does not need any other modification.

4. Discuss the similarities and differences between proxy caching and CDN. Is it beneficial to utilize both of them in a system?

   Answer:

   Both of them replicate content, making them closer to end-users (thereby alleviating the server bottleneck and reducing long-haul traffic) and improving availability. Proxy caching is in general "passive" and "pull-based", in the sense that the proxies are deployed by local users/network operators, and only the content that has been accessed by the clients will be cached. CDN is "proactive" and "push-based", which is deployed globally by CDN operators and replicates content proactively.

   Yes, these two can work together with a better overall performance.

5. Discuss the similarities and differences between a CDN for web content distribution and that for multimedia streaming. What is the role of *reflectors* in Akamai's streaming CDN?

   Answer:

In both cases, the content are replicated to geo-distributed CDN servers to reduce the access latency and long-haul traffic, as well as to improve the content availability.

The web objects are generally of very small sizes (but a massive number). Multimedia objects (audio/video streams) are generally of much larger sizes (but fewer number), and the streaming session can last much longer with diverse interactions (fastforward, backward, random-seek, pause, and etc.). The CDN there faces different scalability challenges.

Reflectors sit between the entrypoints and the edge servers. Each reflector can receive one or more streams from the entrypoints and then send those streams to one or more cluster of edge servers. This enables rapid replicating of a stream to a large number of edge clusters should the streaming event become extremely popular.

6. For Staggered broadcasting, if the division of the bandwidth is equal among all K logical channels ($K \geq 1$), show that the access time is independent of the value of K.

   Answer:
   Let M be the number of movies, L the length of each movie, and B the bandwidth of the link, then the access time for any movie is $\delta = \frac{M \cdot L}{B}$ which is independent from K.

   If $K = 1$, it is a simple round-robin. When $K > 1$, the movies are staggered — customer can get the movie from the next available (logical) channel. However, since the same total bandwidth B now had to be divided among K logical channels, the access time is not improved.

7. Given the available bandwidth of each user, b_1, b_2, ..., b_N, in a multicast session of N users, and the number of replicated video streams, M, develop a solution to allocate the bitrate to each stream, B_i, $i = 1, 2, ..., M$, so that the average *inter-receiver fairness* is maximized. Here, the inter-receiver fairness for user j is defined as $max\frac{B_k}{b_j}$ where $B_k \leq b_j$, $k = 1, 2, ..., M$, i.e., the video stream of the highest rate that user j can receive.

   Answer:

   There can be several different solutions to this problem.

   (a) We can assume that the bitrates of the video streams come from a discrete set. This assumption is valid with any practical video encoders. The rates for the M streams can then be obtained through an exhaustive search. It works when M is small, but the complexity grows exponential with M. This can be improved through dynamic programming, considering that a receiver always subscribes to a stream whose rate is closest but no more than its bandwidth. We can then start from the stream of the lowest rate and allocate the bitrate for each additional stream one by one.

   (b) Alternatively, we can start from an initial rate allocation for the video streams, and then iterative optimize the allocation. Many different iterative optimization algorithms, e.g., clustering algorithms, can be applied in this context.

8. In Receiver-driven Layer Multicast (RLM), why is shared learning necessary? If IntServ or DiffServ is deployed in the network, will RLM still need shared learning?

   Answer:

   With the best-effort Internet, one RLM user's join-experiments can induce packet losses experienced by others sharing the same bottleneck link (as there is not packet prioritization or

flow isolation). These losses would occur frequently if all the users perform uncoordinated join-experiments.

Such problems can be alleviated by IntServ or DiffServ.

9. In a multicast scenario, too many receivers sending feedback to the sender can cause a *feedback implosion* that would block the sender. Suggest two methods to avoid the implosion and yet provide reasonably useful feedback information to the sender.

   Answer:

   Some possible solutions: Randomly-delayed feedback (each receiver randomly delays its feedback; if the sender respond to some earlier feedback, the a receiver may not need to send its own feedback); Localized feedback (the range of the feedback is controlled); Sampled feedback (each receiver sends feedback with certain probability, and the sender operates based on statistics of the feedbacks).

10. To achieve TCP-Friendly Rate Control (TFRC), the Round-Trip Time (RTT) between the sender and the receiver must be estimated (see Chapter 15.3.2). In the unicast TFRC, the sender generally estimates the RTT and hence the TCP-friendly throughput, and accordingly controls the sending rate. In a multicast scenario, who should take care of this and how? Explain your answer.

    Answer:

    This should be taken care of by receivers. First, the RTTs are different for different receivers, and there is no a single standard RTT for all; Second, the receivers can be heterogeneous with different TCP-friendly bandwidth and so there is no a single target transmission rate for the sender (layered video could be used in this case); Third, the feedback collection and RTT estimation, if all done by the sender, cannot scale to large multicast sessions.

11. In this question, we explore the scalability of peer-to-peer, as compared to client/server. We assume that there is one server and N users. The upload bandwidth of the server is S bps, and the download bandwidth of user i is D_i bps, $i = 1, 2, ..., N$. There is a file of size M bits to be distributed from the server to all the users.

    (a) Consider the client/server architecture. Each user is a now a client that is directly served by the server. Calculate the time to distribute the file to all the users.

    (b) Now consider the peer-to-peer architecture. Each user is now a peer, who can either download directly from the server or from other peers. Assume that the upload bandwidth of user i for serving other peers is U_i bps, $i = 1, 2, ..., N$. Calculate the time to distribute the file to all the users.

    (c) Using the results, explain in what conditions will peer-to-peer scale better (with more users) than client/server. Are these conditions naturally satisfied in the Internet?

    Answer:

    (a) $\max\{NM/S, M/\min_i(D_i)\}$

(b) $\max\{M/S, M/\min_i(D_i), NM/(S+\sum_i U_i)\}$

(c) **Compare the items in the above results, if the bottleneck is at $\min_i(D_i)$, then client/server and peer to peer will have similar performance. If the bottleneck is at S, then peer-to-peer will have better performance. In the Internet, we usually have a large number of clients. Note that in client/server, the distribution time increases linearly with increasing client number N. In peer-to-peer, we have additional bandwidth in between clients to reduce the distribution time.**

12. Discuss the similarities and differences between peer-to-peer file sharing and peer-to-peer live streaming. How will such differences affect the implementation of a peer-to-peer living streaming? And how will they affect the calculation in the previous question.

Answer:

Both of them enable peers to directly exchange data blocks. In peer-to-peer file sharing, the peers can exchange any data blocks of the whole file at any time (as long as neighbors need them). Peer-to-peer streaming however has stringent delay constraints and the content of interest updates over time. As such, a sliding window is used, together with delay-aware scheduling for fetching blocks from neighbors. It also has minimum bandwidth requirement to ensure continuous playback.

13. Consider tree-based and mesh-based overlays for peer-to-peer streaming.

(a) Discuss the pros and cons of each of them.

(b) Why is the pull operation used in mesh-based overlays?

(c) Propose a solution (other than those introduced in the book) to combine them toward a hybrid overlay. You may target different application scenarios, e.g., for minimizing delay or for multichannel TV broadcast where some users may frequently change channels.

Answer:

(a) **Tree-based: lower traffic overhead, but cannot fully explore the bandwidth among peers; less robust, needs frequent repair. Mesh-based: more robust; overhead for buffer map exchange; needs sophisticated scheduling algorithm.**

(b) **In a mesh-based overlay, there is no well-defined parent/children relation. Therefore push operation may introduce redundant traffic, i.e., two neighbors push the same data block to a peer;**

(c) **For example, if there are two classes of users: delay-sensitive (with possibly different delay constraints) and insensitive (no constraint), we may organize the first class of users into a tree according to their delay constraints, whereas others are organized through the mesh.**

14. Consider Skype, a popular Voice-over-IP (VoIP) application using peer-to-peer communication. The peers in Skype are organized into a hierarchical overlay network, with the peers being classified as *super peers* or *ordinary peers*, as illustrated in Fig. 16.1. When two Skype users (caller and callee) need to set up a call, both the ordinary peers and the super peers can serve as relays.

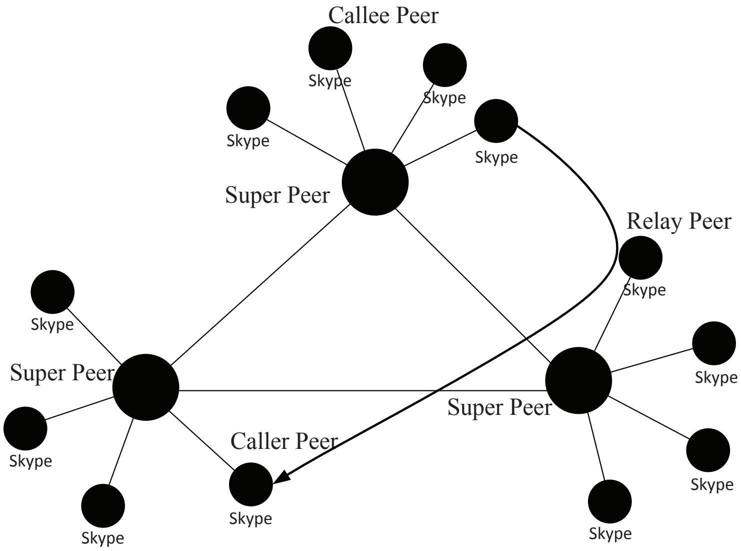


Fig. 16.1: An illustration of the Skype peer-to-peer network.

(a) Skype generally uses UDP for audio streams but TCP for control messages. What kind of control messages are necessary for Skype peers, and why is TCP used?

(b) Explain the benefit of distinguishing super peers and ordinary peers.

(c) Besides one-to-one calls, Skype also supports multi-party conferences. How many copies of audio streams would be delivered in an N user conference, if each user needs to send its copy of stream to all others?

(d) Note that this number can be high. Skype reduces it by asking each user to send its stream to the conference initiator, who will combine the streams into one stream and then forward to each of the other users. How many streams are to be forwarded in the whole conference now? Discuss the pros and cons of this solution, and suggest improvements.

Answer:

(a) Skype maintains an index that maps user names to current IP addresses and port numbers, and the information are exchanged among super peers. Such control information are critical to the operation of Skype, and therefore the reliable TCP is used.

(b) The super peers can serve as relays for peers behind NAT. For example, if both caller and callee are behind NAT, they will have problem in establishing a call directly. In this case, the two peers can connect first to their respective super peers (which are not behind NAT), and later they can make calls through the super peers.

(c) $N(N-1)$

(d) $2(N-1)$**. It greatly saves bandwidth, but the workload of initiator can be very high. There are also other issues like synchronization.**

15. One important reason that HTTP was not traditionally used for media streaming is that the underlying TCP has highly fluctuated transmission rate (the *saw-tooth behavior*), and during severe congestion or channel errors, it may persistently block the data pipe. Explain how DASH addresses these problems. Also discuss other supports for streaming that are missing in the basic HTTP but addressed in DASH.

    Answer:

    The key to support streaming with HTTP is to break the overall media stream into a sequence of small HTTP-based file downloads; each download includes one short chunk of an overall potentially unbounded stream. Using a series of the HTTP's `GET` commands, a user can progressively download the small files while playing those already being downloaded. Any damaged or delayed block will have limited impact (won't block the connection), thus ensuring continuous playback.

    Each client can keep a record of its playback progress, and the progressive download also allows a client to seek to a specific position in the media stream by downloading the corresponding file, or more precisely, performing an HTTP's `byte range` request for the file, realizing similar functionalities offered by RTSP.

Chapter 17

Multimedia over Wireless and Mobile Networks

Exercises

1. In the implementations of TDMA systems such as GSM, an FDMA technology is still in use to divide the allocated carrier spectrum into smaller channels. Why is this necessary?

   Answer:
   It is necessary to divide the allocated spectrum due to device and physical limitations. When the time interval gets too small, the cost for time synchronization can be quite high, and the mutipath fading effect can make the signal interfere with other time intervals at the base station. The electronics also generate thermal noise and can lack the precision to separate signals at very short time intervals. Combining TDMA and FDMA offers a cost-effective solution.

2. We have seen a geometric layout for a cellular network in Figure 17.4. The figure assumes hexagonal cells and a symmetric plan (i.e., that the scheme for splitting the frequency spectrum over different cells is uniform). Also, the reuse factor is $K = 7$. Depending on cell sizes and radio interference, the reuse factor may need to be different. Still requiring hexagonal cells, can all possible reuse factors achieve a symmetric plan? Which ones can? Can you speculate on a formula for general possible reuse factors?

   Answer:
   Not all possible reuse factors can achieve a symmetric plan; for example when the reuse factor is $K = 2$ there is no symmetric plan since one of the cells must border a cell with the same frequencies, while the other does not. The possible reuse factors are 1, 3, 4, 7, 9, 12, ... and the formula is:

   $$K = (i + j)^2 - i \cdot j, \qquad\qquad i, j = 0, 1, 2, 3, ...$$

3. Consider the hard handoff and soft handoff for mobile terminals moving across cells.

   (a) Why is a softer handoff possible with CDMA. Is it possible with TDMA or FDMA?

   (b) Which type of handoff works better with multimedia streaming?

   Hint: During handoff in a CDMA system, the mobile stations can transmit at lower power levels than inside the cell.

   Answer:

(a) **Soft handoff can be realized in CDMA through using different transmission codes on different physical channels, and at lower transmission power levels for outside than inside the cell. When TDMA (or FDMA) is used, given the physical constraints, it is generally hard for a single devices to access two different time slots (frequencies) in separate physical channels.**

(b) **In general, soft handoff works better for multimedia streaming that demands continuous playback.**

4. Most of the schemes for channel allocation discussed in this chapter are fixed (or uniform) channel assignment schemes. It is possible to design a dynamic channel allocation scheme to improve the performance of a cellular network. Suggest such a dynamic channel allocation scheme.

   Answer:
   One idea would be to keep dynamic stats in a centralized way (across many cells) about how many mobiles are trying to access the network at any particular cell at any time. The more mobiles accessing it the more channels should be allocated there and taken away from the least used neighboring cell which can recursively take channels away from its neighbors. Possible algorithms to achieve it can use dynamic programming and Reinforced Learning which is argued to be even more efficient.

5. The Gilbert-Elliot two-state Markov model has been widely used in simulations to characerize wireless errors, as illustrated in Fig. 17.2.

   (a) Given the state transition probabilities p_{00}, p_{11}, p_{10}, p_{01}, calculate steady-state probability P_0 and P_1 that the wireless channel is in state 0 and state 1, respectively.

   (b) Write a simple program to simulate the process. Run it for a long enough time and calculate the average length of error bursts. Discuss how it would affect multimedia data transmission.

   Answer:

   (a) **For state 0: $\frac{p_{10}}{p_{10}+p_{01}}$; For State 1: $\frac{p_{01}}{p_{10}+p_{01}}$.**

   (b) **The simulation process can be divided into a series of time slots; in each slot, based on the transition probabilities, using a random number generator to decide whether to stay in the current state or move to another state. After running for a while, the average results should converge, i.e., won't change much anymore, and we can consider this as a long enough time for simulation.**

6. Consider a wireless network whose signal does not decay dramatically, i.e., within the network range, the signal strength is always high enough. However, the signal can be blocked by physical barriers. Will this network have the hidden-terminal problem? Briefly explain your answer.

   Answer:
   Yes. For example, if the signal between A and B is blocked, but the signal between A and C or B and C is not, then A and B are hidden to each other.

7. In today's networks, both the transport layer and link layer implement error detection mechanisms. Why do we still need error detection in the link layer given that the transport layer protocol, say TCP, assumes that the lower layers of a network is unreliable and seeks to guarantee reliable data transfer using error detection and retransmission? Hint: Consider the performance gain.

Answer:
TCP in the transport layer is end-to-end, which detects an error only after the packet reaches the receiver (after possibly many hops beyond where the error occurs). Error detection in the link layer however can localize the error much more quickly and earlier. Therefore, even a slight improvement in the link layer (not necessarily 100% reliable) can greatly improve the overall performance.

8. Discuss the error detection and correction capability of the two-dimensional parity check.

   Answer:
   It can detect and correct any single bit error, as well as detect any two bit errors (not always correctable though).

9. Calculate the Internet checksum of the following message: 10101101 01100001 10001000 11000001

   Answer:
   1100100111011100

10. Consider Cyclic Redundancy Check (CRC).

    (a) Assume the key word, K, is 1001, and the message M is 10101110. What is the width (in bits) of the CRC bits, R? What is the value of R? Please give detailed calculations.

    (b) Prove that $M \cdot 2^r \oplus R$ is perfectly divisible by K, and verify it using the M, K, and R values above.

    Answer:

    (a) CRC bits $= 001$**; width is** 3

    (b) $M \cdot 2^r = n \cdot K \oplus R$ **for some** n**, which follows** $M \cdot 2^r \oplus R = n \cdot K$**. All these follow the modulo-2 arithmetic.**

    For $M = 10101110$, $K = 1001$**, and** $R = 001$**, we have that 10101110001 is perfectly divisible by 1001.**

11. Discuss why interleaving increases the delay in decoding? Will interleaving be effective if the loss is uniformly distributed?

    Answer:
    For a packet that is interleaved into n **packets, only after all the** n **packets are received will the packet be recoverable. This introduces delay, particularly for the earlier packets in a group. Interleaving is less effective with uniform loss.**

12. H.263+ and MPEG-4 use RVLCs, which allow decoding of a stream in both forward and backward directions from a synchronization marker.

    (a) Why is decoding from both directions preferred?

    (b) Why is this beneficial for transmissions over wireless channels?

    (c) What condition is necessary for the codes to be reversibly decodable? Are these two set of codes reversible: (00,01,11,1010,10010) and (00,01,10,111,110)?

    (d) Why are RVLCs usually applied only to motion vectors?

Answer:

(a) **With the conventional VLC, a single-bit error can cause continuous errors in reconstructing the data even if no further bit error happens. In other words, the information carried by the remaining correct bits become useless. If we can decode from the reverse direction, then such information could be recovered. Another potential use of RVLC is in the random access of a coded stream. The ability to decode and search in two directions should halve the amount of indexing overhead with the same average search time as compared to the standard one-directional VLC.**

(b) **Over wireless channels there is noise that can produce random bit errors, and we would like to avoid retransmission as bandwidth is more limited and most applications are real time. It is important then to use RVLC at a slight expense of coding efficiency.**

Only the first set of codes is reversible.

(c) **RVLCs are not applied to DCT coefficient since these are viewed as much less important if lost and easier to estimate than MVs. MVs are essential at most frames for reasonable visual quality, the residuals are not as important, and although I-frames can be treated differently, there aren't that many I-frames in a video sequence comparatively.**

13. Suggest two error concealment methods for audio streaming over wireless channels.

Answer:

Some possible solutions: Repeat the previous sample; using interpolation to recover the sample; convert to the tansform domain, perform smoothing, and then convert back to the time domain.

14. There is a broad spectrum of device and user mobility, in terms of both range and speed, as illustrated in Figure 17.14. Discuss the challenges in the different mobility scenarios, and the potential solutions.

Answer:

- ***Micro-mobility*** **(intra-subnet mobility), where movement is within a subnet. No special mechanism is needed as long as the physical and datalink layers can maintain the communication between the base station and the mobile terminal.**
- ***Macro-mobility*** **(intra-domain mobility), where movement is across different subnets within a single domain. Handoff management in the link layer and mobile IP in the network layer are to be used.**
- ***Global mobility*** **(inter-domain mobility), where movement is across different domains in various geographical regions. Mobile IP and location/roaming management are to be used.**

15. To alleviate triangular routing, a CN can also keep the mapping between the mobiles HoA and CoA, and accordingly encapsulate packets to the mobile directly, without going through the HA.

(a) In which scenario does this *direct routing* solution work best?

(b) Discuss any potential problem with the direct routing solution.

(c) Propose another solution that addresses the triangular routing problem. Discuss its pros and cons.

Answer:

(a) If the mobile node moves infrequently (stays in a visited network for a relatively long time once moves there) and the file to be transferred is huge, direct routing works best.

(b) If the mobile nodes moves frequently across networks, then the communication with a correspondent can be easily broken.

(c) One possible solution is to use a cross-layer design. The upper layer (transport or application layer) may predict the mobility pattern and accordingly inform the network layer to use direct or indirect routing. Buffers in the application layer can also be used to pre-fetch data in the case of movement, so as to avoid data outage.

Chapter 18

Social Media Sharing

Exercises

1. Find out a typical Web 1.0 application and a typical Web 2.0 application, and discuss their key differences.

   Answer:
   Examples:

   (a) Web 1.0: a plain HTML-based webportal; users passively browse the content.

   (b) Web 2.0: Facebook, Twitter, YouTube, with much richer user interactions and user generated content.

2. Discuss the key differences between YouTube videos and the traditional movies/TV shows. How would they affect content distribution?

   Answer:

   YouTube videos are mostly generated by generic users. These videos are relatively short and of large quantity, but the quality control is loose. Traditional movies/TV shows are generated by professional content producers with good quality control, generally longer, and of smaller quantity.

3. YouTube publishes statistics about its videos online. As of the end of 2013, we have the following statistics:

   - More than 1 billion unique users visit YouTube each month
   - Over 6 billion hours of video are watched each month on YouTube
   - 100 hours of video are uploaded to YouTube every minute
   - 80% of YouTube traffic comes from outside the US
   - YouTube is localized in 61 countries and across 61 languages
   - Mobile makes up almost 40% of YouTube's global watch time

   Check the recent statistics and estimate the monthly growth speed of YouTube. Suggest some reasons that make YouTube-like services expand so quickly and the potential challenges therein.

Answer:

The statistics can be found from YouTube's webpage.

4. Is it beneficial to place all the content from a social media service in one server? If not, what are the challenges to place the content in multiple servers?

   Answer:

   Not necessary if the content is large and the users are geo-distributed.

   Placing the social media content to multiple servers needs to meet two objectives (a) load balancing; (b) preserving social relations (content of socially close users should be placed together), and these two objectives however can conflict with each other.

5. Discuss the propagation and consumption patterns of multimedia content in a social networking tool that you're familiar with.

6. Given a positive integer n and a probability value $0 \leq p \leq 1$, an *Erdös-Rényi* (ER) random graph $G(n, p)$ is with n vertices where each possible edge has probability p of existing. This is the most important class of random graphs.

   (a) Write a simple program to generate ER random graphs, and calculate their characteristic path lengths and clustering coefficients. Compare them with the YouTube video graph we have discussed earlier.

   (b) Discuss whether the graph formed by an online social network, say the graph of Facebook user accounts, is such a random graph or not. Hint: Think about the way that the edges are formed.

   Answer:

   (a) You will find that the YouTube video graph has much shorter characteristic path length and much higher clustering coefficients.

   (b) In general, not a random graph. One explanation is given by the so-called "preferential attachment"; that is, users tend to attach to popular users, making them even more popular.

7. A simple model for information propagation is *gossip*. With gossip, a network node, upon receiving a message, will randomly forward it to the neighboring nodes with probability p.

   (a) Write a simple program to simulate the gossip algorithm in randomly generated networks. A node may delay t time before forwarding. Discuss the impact of p and t in the coverage and propagation speed of a message.

   (b) Is it beneficial if the nodes can have different values of p? If so, provide some guidelines in the selection of p for each node.

   (c) Is gossip suitable for modeling the propagation process of a picture shared in a realworld social network, say Facebook? How about video ?

   Answer:

(a) **The implementation can be discrete event driven, that is, dividing the process into a series of time slots, and for each slot, calculating the forwarding behavior for each node based on p.**

(b) **It would be beneficial if the graph is not regular. For example, if two partitions of a graph are connected by a single link, then p of its two nodes should be set to 1 so as to ensure that the messages can be propagated from one part of the graph to the other; on the other hand, if a node's in-degree and out-degree are both very high, then its p value should be set to low.**

(c) **For pushed content, gossip can serve as a basic model for the propagation process, although p depends on individual users or content. For video, the basic gossip is not a complete model as the video content in general is not pushed — only the video link (and some summary) is the pushed; the recipient needs to first decide whether to watch the video and then whether to forward it.**

8. In an online social network, a free rider only consumes videos but does not share videos. Free riders also exist in peer-to-peer file sharing: they download data segments from others, but never upload. BitTorrent adopts a *tit-for-tat* strategy to solve the incentive problem, i.e., you get paid only if you contribute. As depicted in Fig. 18.1, peers A, B, and C download different segments from each other. This forms a feedback loop; for example, uploading segment 2 from A to B will be feedback to A by the upload of segment 3 from C to A, which stimulates peer A to cooperate.

   (a) Discuss whether the tit-for-tat strategy works for video propagation with free riders.

   (b) For live video streaming with delay constraints, with tit-for-tat work?

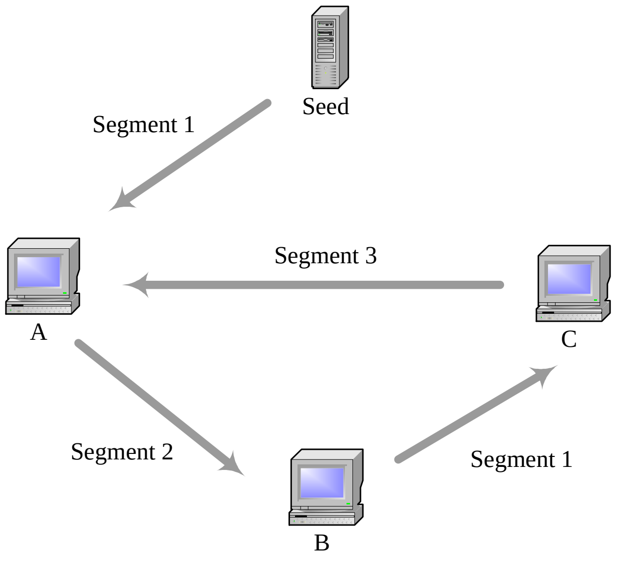


Fig. 18.1: An example of the tit-for-tat strategy.

Answer:

(a) **This depends on the purpose of video propagation in the social network. If it is to ensure that there is no free rider, then tit-for-tat is a good incentive mechanism. However, if it is to**

propagate videos to as many users as possible, then there is no need to use tit-for-tat as it may confine the propagation range (This differs from the BitTorrent scenario, where fair and efficient sharing is the most important requirement).

(b) Tit-for-tat doesn't work well in this case. Hint: The peers may join the session at different times. For file sharing, any peer is interested in the whole file, no matter it joins earlier or later. Yet for live streaming, a newly joined peer will not be able to access earlier streaming. As such, the credits accumulated by different peers are not equivalently useable.

9. The basic binary tree in COOLS can be quite high. For example, when there are 1000 nodes, the tree height can easily reach to 10.

   (a) What are the potential problems with a tall tree.

   (b) One simple solution to reduce the height of the tree to is increase the number of children for each node. Will this solution work for COOLS?

   (c) Suggest a possible solution that practically works and analyze its effectiveness.

   Answer:

   (a) A tall tree can be more vulnerable to node dynamics, and the nodes at the bottom of the tree will suffer from longer delays;

   (b) The number of nodes in each depth is growing linearly. Hence, simply increasing the number of children for each node would remarkably increase the load of the nodes close to the root.

   (c) One possible solution: Let the root node has 2^k children $(k \geq 0)$, then the nodes at depth i have at most 2^{k-i} children, and the tree height is no greater than $k+1$. Given that a complete tree in which the root node has 2^k children, there are 2^k nodes at depth 1, and $2^k \cdot 2^{k-1}$ nodes at depth 2, and so on. Thus the number of nodes at depth i is at most

$$N_i = \prod_{j=1}^{i} 2^{k-j+1}.$$

   And the total number of nodes (excluding root) in the tree can be calculated as

$$\sum_{i=1}^{k} N_i.$$

   Since the number of nodes at depth i is $\prod_{j=1}^{i} 2^{k-j+1}$, and that at depth $(i-1)$ is then $\prod_{j=1}^{i-1} 2^{k-j+1}$, and thus the growth factor at depth i is 2^{k-i+1}. As i increases, the factor decreases, and thus the number of nodes at each depth is increasing sub-linearly.

Chapter 19

Cloud Computing for Multimedia Services

Exercises

1. Discuss the relations and differences of the following systems: Cloud, Server Cluster, Content Distribution Network (CDN), and Datacenter.

   Answer:

   Cloud computing relies on sharing of resources to achieve coherence and economies of scale. Cloud users can run their applications on powerful server clusters offered by the cloud service provider, with system and development software readily deployed inside the cloud. A datacenter is a dedicated facility to house large server clusters.

   A CDN is mainly used for efficient content distribution through geo-distributed servers. Many cloud applications use a CDN for content distribution, e.g., NetFlix.

2. Consider cloud-based video streaming and peer-to-peer video streaming

   (a) For each of them, what are the pros and cons?

   (b) Discuss a possible solution that combines these two. Discuss the benefit of this hybrid design and its potential issues.

   Answer:

   (a) The always-on and centralized data centers can make one-to-many sharing highly efficient and synchronous – uploaded content can be readily shared to a large population instantly or lately. Sharing through a cloud could also offer better QoS given that the connections with data-centers are generally good, not to mention the firewall and network address translation traversal problems commonly encountered in peer-to-peer sharing. It also allows designers to start a service small but easy to scale large.

   Peer-to-peer can be even cheaper and scalable. It however can hardly guarantee QoS with dynamic peers and the startup delay can be long. When the peer-to-peer overlay is very small, i.e., for a non-popular video, it may not function well. The dramatically increased traffic, obstacles created by NAT/firewalls, and copyright infringement are also concerns.

(b) **One possible solution is to use cloud to serve the users of non-popular videos or the users of poor connections (possibly behind NAT/firewall), and then organize powerful peers of popular videos into peer-to-peer overlays for scalable content sharing. In the case of peer departures or failures, or flash crowd, the elastic resources in the cloud can serve any peers suffering from such events.**

Besides the increase complexity for implementing the hybrid design, some inherent issues of peer-to-peer, e.g., copyright infringement, remain to be addressed in this hybrid design.

3. Consider the cloud-based Netflix Video-on-Demand (VoD) service.

(a) Describe the respective roles of Amazon EC2 and S3 in Netflix.

(b) Netflix uploads the master videos to the cloud for transcoding. Why doesn't Netflix transcode the videos locally and then upload them to the cloud?

(c) Why does Netflix still need a CDN service beyond S3?

Answer:

(a) **Amazon EC2 is mainly used for *Content Conversion*. Netflix purchases master copies of digital films from movie studios and, using the powerful EC2 cloud machines, converts them to over 50 different versions with different video resolutions and audio quality, targeting a diverse array of client video players running on desktop computers, smartphones, and even DVD players or game consoles connected to television;**
S3 is used for *Content Storage*. The master copies and the many converted copies are stored in S3.

(b) **The transcoding is both computation- and data-intensive, which requires a high performance computing infrastructure. Instead of maintaining such an expensive infrastructure and expands its scale over time (or reduces if the service scale shrinks), Netflix has completely migrated its services to the Amazon's cloud, which provides scalable, elastic and high-performance computing resources for the transcoding. Also consider today's relatively low costs for bandwidth, uploading the master videos to the cloud and then transcoding won't suffer much from the bandwidth cost than transcoding locally and then uploading.**

(c) **S3 is used for content storage. It provides scalable and highly reliable storage. However, the Netflix users are distributed worldwide, and as a paid service, Netflix needs offer stable streaming quality and low startup delay. Therefore, a CDN is still needed beyond S3 storage.**

4. Is it always beneficial to offload computation to the cloud? List two application scenarios that simple computation offloading may not be cost-effective, and suggest possible solutions.

Answer:

(a) **Not always.**

(b) **An example is video compression, whose data have very high volume and dependency; simple offloading will require the uploading of the whole raw video, which is not cost-effective. A solution is to use Cloud-Assisted Motion Estimation (CAME) that uses mesh-based motion estimation and offloads it to cloud.**

Another example is file storage in the cloud. Compressing the files can save the storage space in the cloud. Depending on the bandwidth cost and energy consumption of network interfaces and the local CPU, local compression would be better than uploading the raw file and then compressing in the cloud (say for a mobile user). See Question 6 for detailed calculation.

5. In this question, we try to quantify the cost savings of using cloud services. Without the cloud, a user has to purchase his or her own PC, say of price $\$X$. The value of the machine depreciates at a rate of $p\%$ per month, and when the value is below $V\%$, the machine is considered out-dated and the user has to purchase a new machine. On the other hand, using the cloud, the user doesn't have to buy his or her own machine, but leases from the cloud service provider with a monthly fee of $\$C$.

   (a) To make the cloud service cost-effective, how much should the provider set for the monthly lease fee $\$C$ for a comparable cloud machine instance?

   (b) Are there any other realworld costs that can be included in the model, associated either with local machine or with the cloud?

   Answer:

   (a) The real monthly value of machine deprecates is $\frac{X \cdot p\%}{(100-V)\%}$. Hence, the monthly lease fee C should be less than $\frac{X \cdot p\%}{(100-V)\%}$.

   (b) Other costs with local machines: electricity costs, cooling costs, the costs of high bandwidth plan if need to support bandwidth-intensive services, and etc. With cloud: a lightweight PC for the user to connect to the cloud, a network plan, etc.

6. Considering the energy consumption of a task with local computing at a mobile terminal and with offloading to the cloud. We assume the task needs C CPU cycles for computation. Let M and S be the computing speeds, in CPU cycles per second, of the mobile terminal and of the cloud machine, respectively. The local computing at the mobile terminal has energy consumption of P_M watts, and incurs no data exchange over the wireless interface. For offloading to the cloud, D bytes of data are to be exchanged over the wireless interface. We assume that the network bandwidth is B bps and the energy consumption of the air interface during transmitting or receiving is P_T watts.

   (a) Assume that the CPU of the mobile terminal consumes no energy when it is idle, nor does the wireless interface of the terminal. What is the energy consumption of the mobile terminal if the task is executed locally, or offloaded to the cloud? Note that we don't consider the energy consumption in the cloud because the energy bottleneck of interest here is at the mobile terminal.

   (b) Under what condition does offloading to the cloud save energy?

   (c) What are other potential benefits with computation offloading, and under what conditions?

   Answer:

   (a) Energy consumption of local execution: $P_M \times \frac{C}{M}$; Energy consumption of offloading: $P_T \times \frac{8D}{B}$.

(b) Saving energy requires $P_T \times \frac{8D}{B} < P_M \times \frac{C}{M}$. Since P_M, P_T, B, M are constants, C should be large and D should be small, which means that applications with high computation demand and minimal data transmission if offloaded to the cloud are suitable for offloading to save energy.

(c) Tasks can be completed sooner if $\frac{8D}{B} + \frac{C}{S} < \frac{C}{M}$. With computation offloading, low-end mobile terminals can conduct complex tasks. This needs ubiquitous connection to Internet.

7. Besides the cost or energy savings, list two other advantages when using a cloud, as compared to building and maintaining a local infrastructure. Also list two disadvantages.

Answer:

(a) Anywhere and anytime access; Elastic capacity (easily accommodate bursty requests in a popular event); Resistent to malicious attacks (major cloud datacenters are much better protected than the computing infrastructure for small business).

(b) High demand on networking (traffic/reliability/availability); Security concerns (during uploading/downloading, and for the content stored in remote).

8. Consider cloud gaming, in which game scenes are rendered in the cloud and then streamed back to a thin client.

(a) What are the benefits of using a cloud for gaming?

(b) What types of games are most suitable for cloud gaming?

(c) Discuss the requirements for live video streaming and those for cloud gaming. How are they similar? What special requirements of cloud gaming make it more difficult?

(d) Suggest some solutions that can reduce the delay in cloud gaming.

Answer:

(a) Cloud gaming can bring great benefits by expanding the user base to the vast number of less-powerful devices that support thin clients only, particularly smartphones and tablets. As such, mobile users can enjoy high-quality video games without performing the computation-intensive image rendering locally. It further reduces customer support costs since the computational hardware is now under the cloud gaming provider's full control, and offers better Digital Rights Management (DRM) since the codes are not directly executed on a customer's local device.

(b) Both cloud gaming and live video streaming must quickly encode/compress incoming video and distribute it to end users. In both cases, only a small set of the most recent video frames are of interest, and there is no need or possibility to access future frames before they are produced, implying encoding must be done with respect to very few frames.

Yet conventional live video streaming and cloud gaming have important differences. First, compared to live video streaming, cloud gaming has virtually no capacity to buffer video frames on the client side. Live video streaming on the other hand can afford a buffer of hundreds of milliseconds or even a few seconds with very little loss to the QoE of the end user.

(c) Some possible solutions: (a) smart partitioning of the jobs between local hardware and the remote cloud; (b) smart reference scheme for video compression.

Chapter 20

Content-Based Retrieval in Digital Libraries

Exercises

1. Devise a text-annotation taxonomy (categorization) for image descriptions, starting your classification using the set of Yahoo! categories, say.

   Answer:
   Yahoo has a top-level taxonomy of 13 classes, such as Art, Entertainment, Finance, News, Sports, etc. Within sports.yahoo.com, there is a sub-taxonomy of a further 9 classes (Baseball, Basketball, Football, Hockey, Soccer, MoreSports, Sailing, OtherSports, Pick'emGames), and of course a finer classification within these.

2. Examine several web site image captions. How useful would you say the textual data is as a cue for identifying image contents? (Typically, search systems use *word stemming*, for eliminating tense, case, and number from words — the word *stemming* becomes the word *stem*.)

   Answer:
   The caption is most important in combination with the image contents:

   > **"The period following the split announcement, the Post-Announcement stage, often sees a *depression*."**

   However, in *"Construction of a Hierarchical Classifier Schema using a Combination of Text-Based and Image-Based Approaches", Cheng Lu and Mark S. Drew, ACM SIGIR 2001, The 24th Annual Conference on Research and Development in Information Retrieval, pp.438-439, New Orleans, September 9-13, 2001,*
   http://www.cs.sfu.ca/∼mark/ftp/Sigir01/sigir01.pdf
   we showed that performing classification on a hierarchy differently on different levels of the tree, using text for branches and images only at leaves, improved web document classification performance significantly.

3. Suppose a color histogram is defined coarsely, with bins quantized to 8 bits, with 3 bits for each red and green and 2 for blue. Set up an appropriate structure for such a histogram, and fill it from some image you read. Template Visual C++ code for reading an image is on the text web site, as `sampleCcode.zip` under "Sample Code".

Answer:
Here we shall simply give a `matlab` **version:**

```
% rgbhist.m
% matlab script
im=imread(’resources_exercises/chap18/lena256.jpg’);
ss = size(im); % rows x cols x 3
% let’s just truncate values to 3-bits,3-bits,2-bits for R,G,B:
im = double(im);
imR = im(:,:,1);
imG = im(:,:,2);
imB = im(:,:,3);
scaleR = 256/8; % 3-bit
scaleG = 256/8; % 3-bit
scaleB = 256/4; % 2-bit
imR =  fix(imR/scaleR)*scaleR;
imG =  fix(imG/scaleG)*scaleG;
imB =  fix(imB/scaleB)*scaleB;
im(:,:,1) = imR;
im(:,:,2) = imG;
im(:,:,3) = imB;
%
values8 = unique(fix(0:255/scaleR)*scaleR);
% 0 32 64 96 128 160 192 224
values4 = unique(fix(0:255/scaleB)*scaleB);
% 0 64 128 192
%
im = reshape(im, ss(1)*ss(2), 3);
hist332 = zeros(8,8,4);
for i=1:8
  for j=1:8
    for k=1:4
      temp =  (im(:,1)==values8(i)) & ...
        (im(:,2)==values8(j)) & ...
        (im(:,3)==values4(k));
      % size(temp) = ss(1)*ss(2)
      hist332(i,j,k) = sum(temp);
    end
  end
end
sum(sum(sum(hist332))) % ==ss(1)*ss(2)==number of pixels.
plot3(hist332(:,:,1),hist332(:,:,2),hist332(:,:,3),’.’);
```

4. Try creating a texture histogram as described in Section 18.2.5. You could try a small image and follow the steps given there, using MATLAB, say, for ease of visualization.

   Answer:
   Here we simply give the beginning steps: creation of a good edge map.

```
% texhist.m
% matlab script
im = imread('resources_exercises/chap18/lena256.jpg');
im = rgb2ycbcr(im);
im = im(:,:,1); % luminance
im = double(im);
ss = size(im);
rows = ss(1);
cols = ss(2);

% edge magnitude
sobelx = [[-1 0 1] ;
          [-2 0 2] ;
          [-1 0 1]];
sobely = sobelx';
imdx = conv2(im,sobelx,'same');
imdy = conv2(im,sobely,'same');
D = sqrt(imdx.^2 + imdy.^2);
% edge direction
phi = atan2(imdy,imdx); % in -pi .. pi
phi = phi*180/pi; % in -180 .. 180
codes = direction_code(im);

% non-maximum suppression:
mask = ones(ss); % edge pixels not to discard
% first, don't use border pixels:
mask(:,1) = 0; mask(:,end) = 0; mask(1,:) = 0; mask(end,:) = 0;
for i=2:(rows-1) % avoid borders
  for j=2:(cols-1)
    switch codes(i,j)
      case 0  % horiz
  i1 = i; j1 = j-1; i2 = i; j2 = j+1;
      case 1  % centered on 45 degrees
  i1 = i+1; j1 = j-1; i2 = i-1; j2 = j+1;
      case 2  % vertical
  i1 = i-1; j1 = j; i2 = i+1; j2 = j;
      case 3  % centered on 135 degrees
  i1 = i-1; j1 = j-1; i2 = i+1; j2 = j+1;
    end
    %
    mag = D(i,j);
   % has larger neighbor pixel close in direction
    if ~( ...
   ( mag < D(i1,j1) ) && (~largeanglediff(phi(i,j),
   phi(i1,j1)) ) ...
         ||  ...
         ( mag < D(i2,j2) ) && (~largeanglediff(phi(i,j),
          phi(i2,j2)) ) ...
```

```
        )
      % suppress, since not (local max And same direction):
      mask(i,j) = 0;
    end
  end
end

% mask is local-maximum pixels And D>Threshold
% Choose Thresh as excluding bottom 10% of D values:
[counts,bincenters] = hist(D(:));
thresh = mean(bincenters(1:2));
temp = (D(:)>thresh);
mask2 = reshape(temp,ss);
mask = mask & mask2;

%remove isolated pixels: if 3x3 around pixel is surrounded by ~mask:
mask3 = ones(ss);
for i=3:(rows-2)
  for j=3:(cols-2)
    if (     mask(i,j) &&  ...
              ...
             ~mask((i-2),(j-2)) && ...
             ~mask((i-2),(j-1)) && ...
             ~mask((i-2),j) && ...
             ~mask((i-2),(j+1)) && ...
             ~mask((i-2),(j+2)) && ...
              ...
             ~mask((i-1),(j-2)) && ...
             ~mask((i-1),(j+2)) && ...
              ...
             ~mask(i,(j-2)) && ...
             ~mask(i,(j+2)) && ...
              ...
             ~mask((i+1),(j-2)) && ...
             ~mask((i+1),(j+2)) && ...
              ...
             ~mask((i+2),(j-2)) && ...
             ~mask((i+2),(j-1)) && ...
             ~mask((i+2),j) && ...
             ~mask((i+2),(j+1)) && ...
             ~mask((i+2),(j+2)) ...
       )
       mask3(i,j) = 0;
    end % if
  end
end

mask = mask & mask3;
```

```
myimwrite1(mask+0.0,'mask.jpg');

D = D.*mask;
%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% largeanglediff.m
function bool = largeanglediff(a1,a2)
% more than 45 degrees is "large" difference
temp = abs(a1-a2);
if temp>180
  temp = 360-temp;
end
if temp>45
  bool = true;
else
  bool = false;
end


% direction_code.m
function code = direction_code(angle)
% Make angle into an int==code for four directions,
% centered on 0, 45, 90, 135,
% independent of vector sense.
if angle > 180
  angle = 360 - angle;
end
code = mod( fix((angle+22)/45.0) , 4 );
```

The result of the above, for image `lena` **is shown in Fig. 20.21.**

5. Describe how you may find an image containing some 2D "brick pattern" in an image database, assuming the color of the "brick" is yellow and the color of the "gaps" is blue. (Make sure you discuss the limitations of your method and the possible improvements.)

   (a) Use color only.

   (b) Use edge-based texture measures only.

   (c) Use color, texture and shape.

   Answer:
   Follow the descriptions of C-BIRD in the text, for this task.

6. The main difference between a static image and video is the availability of motion in the latter. One important part of CBR from video is motion estimation (e.g., the direction and speed of any move-

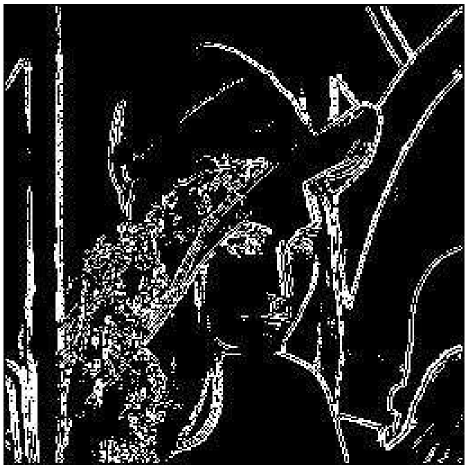


Fig. 20.21: Edge mask.

ment). Describe how you could estimate the movement of an object in a video clip, say a car, if MPEG (instead of uncompressed) video is used.

Answer:
Since MPEG stores motion vector information (in compressed form), motion information is already readily available in the video without further processing being necessary. A simple approach to motion estimation is to adopt the assumption that most parts of the frame are relatively unchanging. Thresholding for higher values of the lengths of motion vectors gives a distribution of motion vectors for the moving object, and these can be used then to estimate a representative overall motion.

If we wish to also take into account the "dominant motion" consisting mostly of camera motion, we can first carry out motion compensation by mapping subsequent frames back in time; here, we assume that most of the frame does not consist of moving objects, so that motion vectors mainly arise from camera zooming, panning, etc. Then the same thresholding of motion-compensated frames and dominant-motion-compensated motion vectors can find the moving object.

7. Color is three-dimensional, as Newton pointed out. In general, we have made use of several different color spaces, all of which have some kind of brightness axis, plus two intrinsic-color axes.

   Let's use a *chromaticity* 2-dimensional space, as defined in Eq. (4.7). We'll use just the first two dimensions, $\{x, y\}$. Devise a 2D color histogram for a few images, and find their histogram intersections. Compare image similarity measures with those derived using a 3D color histogram, comparing over several different color resolutions. Is it worth keeping all three dimensions, generally?

   Answer:
   A few examples will suffice:

```
% matlab script
im = imread('resources_exercises/chap18/lena256.jpg');
ss = size(im);
im = reshape(im,prod(ss(1:2)),3);
ch = makechrom(im);
chshow(ch,ss(1),ss(2));
```

```
chhist = myhist2d(ch); %16 x 16 histogram
mesh(chhist);

% Let’s show that cutting up the image makes no difference:
% flip top and bottom halves:
im = reshape(im,ss);
im2 = im; % declare
im2(1:(ss(1)/2),:,:) = im((ss(1)/2+1):end,:,:);
im2((ss(1)/2+1):end,:,:) = im(1:(ss(1)/2),:,:);
im = reshape(im,prod(ss(1:2)),3);
im2 = reshape(im2,prod(ss(1:2)),3);
ch2 = makechrom(im2);
chhist2 = myhist2d(ch2);
max(max(chhist2-chhist)) %  0


% Let’s just make an image using a diagonal transform:
im = double(im);
im2 = im*diag([0.9, 1.1, 1.05 ]);
ch2 = makechrom(im2);
chshow(ch2,ss(1),ss(2));
chhist2 = myhist2d(ch2);
mesh(chhist2);
% histogram intersection:
%    closer image is neaer to 1.0 value.
intersec = histint2d(chhist,chhist2) % 0.7818
%
% and compare to a 3D hist:
hist3d = myhist3d(im);
hist3d2 = myhist3d(im2);
intersec = histint3d(hist3d,hist3d2) % 0.3527
% so 2D is considerably better than 3D.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%
% makechrom.m
function r=makechrom(mat)
% mat is Nx3
[rows,cols] = size(mat);
mat = double(mat);
denom = mat(:,1)+mat(:,2)+mat(:,3);
back = denom==0;
denom(back)=999;
r = zeros(rows,2);
for k=1:2
 tempr = mat(:,k)./denom;
 tempr(back)=0;
 r(:,k) = tempr;
end

%%%%%%%
% myimshow3.m
function fun(im3,r,c)
im3 = double(im3);
im3 = im3/max(max(im3));
temp=reshape(im3,r,c,3);
imshow(temp);

%%%%%%%
% chshow.m
function fun(ch,r,c)
% param is mxn x 2
%
% make a 3-d chrom, for display
ch3 = [ ch 1-ch(:,1) - ch(:,2) ];
myimshow3(ch3,r,c)


%%%%%%%
% myhist2d.m
function ahist = myhist2d(ch)
%param is mxn by 2; N is 16, say
  N = 16;
  ch = double(ch);
  ch = 255/max(max(ch))*ch;
  ahist = zeros(N,N) ;
  % Just take blacks as all-0 chs:
  blacks = (ch(:,1)==0) & (ch(:,1)==0);
  temp = ch(˜blacks,:);
  % scale to 0..15 and truncate:
  scale = 15/255;
  temp = floor(temp*scale + 0.5)+1; % 1..16
  tempsize = size(temp);
  for rr = 1:tempsize(1)
    ahist(temp(rr,1),temp(rr,2)) =  ...
    ahist(temp(rr,1),temp(rr,2)) + 1;
  end;
  % Now make volume == 1:
  ahist = ahist/sum(sum(ahist));
  %return(ahist)
% end of myhist2d

%%%%%%%
% histint2d.m
```

```
function intersec = histint2d(chhist1,chhist2)
% first, normalize the hist's:
chhist1 = chhist1/sum(sum(chhist1));
chhist2 = chhist2/sum(sum(chhist2));
% hist intersection is sum of min values:
chhist1smaller = (chhist1<chhist2);
minhist = chhist1; % declare
minhist(chhist1smaller) = chhist1(chhist1smaller);
minhist(˜chhist1smaller) = chhist2(˜chhist1smaller);
intersec = sum(sum(minhist));


%%%%%%%
% myhist3d.m
function ahist = myhist3d(im)
%param is mxn by 3; N is 16, say
  N = 16;
  im = double(im);
  im = 255/max(max(max(im)))*im;
  ahist = zeros(N,N,N) ;
  % scale to 0..15 and truncate:
  scale = 15/255;
  im = floor(im*scale + 0.5)+1; % 1..16
  imsize = size(im);
  for rr = 1:imsize(1)
    ahist(im(rr,1),im(rr,2),im(rr,3)) =  ...
    ahist(im(rr,1),im(rr,2),im(rr,3)) + 1;
  end;
  % Now make volume == 1:
  ahist = ahist/sum(sum(sum(ahist)));
  %return(ahist)
% end of myhist3d

%%%%%%%
% histint3d.m
function intersec = histint2d(hist3d1,hist3d2)
% first, normalize the hist's:
hist3d1 = hist3d1/sum(sum(sum(hist3d1)));
hist3d2 = hist3d2/sum(sum(sum(hist3d2)));
% hist intersection is sum of min values:
hist3d1smaller = (hist3d1<hist3d2);
minhist = hist3d1; % declare
minhist(hist3d1smaller) = hist3d1(hist3d1smaller);
minhist(˜hist3d1smaller) = hist3d2(˜hist3d1smaller);
intersec = sum(sum(sum(minhist)));
```

8. Suggest at least three ways in which audio analysis can assist in video retrieval-system-related tasks.

   Answer:
   Firstly, we note that while the video in a scene can contain significant alterations, even cuts, while a scene is playing out, usually the audio remains smooth throughout. That is, one of the salient features of a *scene* is that audio is smooth, even if video is not.

   Secondly, important aspects of a human's speaking voice can be utilized. For example, in a lecture, usually the frequency raises at the beginning of a new topic.

   Thirdly, the characteristics of human speech itself can be used to determine whether people are present in a scene, along with color and other video features.

9. Implement an image search engine using low-level image features such as color histogram, color moments, and texture. Construct an image database that contains at least 500 images from at least 10 different categories. Perform retrieval tasks using a single low-level feature as well as a combination of features. Which feature combination gives the best retrieval results, in terms of both Precision and Recall, for each category of images?

   Answer:
   Simple features such as color histogram (or chromaticity histogram) are included in the textbook website, as the Visual Basic project VbColorIndexingSRC.zip under Student Projects > Color Indexing.

   As for the feature combination, it is probably domain dependent, i.e., no combination will work well for all categories of images.

10. Another way of combining Precision and Recall is the F-score measure. The F-score is the harmonic mean of Precision P and Recall R, defined as

$$F \;=\; 2(P * R)/(P + R)$$

   Experiment and determine how F behaves as P and R change.

   Answer:
   The following figure (Fig. 20.22) tells the story.

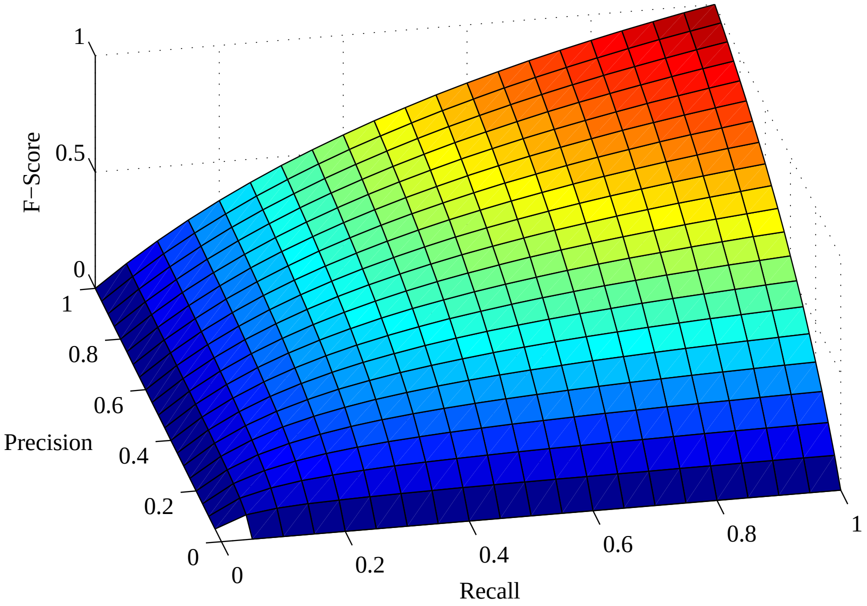


Fig. 20.22: F-score vs. Precision and Recall.